MASTER EN SCIENCES MATHÉMATIQUES

Etude d'un algorithme de minimisation globale en programmation d.c..

Delaunois, Anne; Ternet, Marie-Pascale

Award date:
1988

Awarding institution:
Universite de Namur

FACULTES UNIVERSITAIRES N.D. DE LA PAIX NAMUR

FACULTE DES SCIENCES

Etude d'un algorithme de minimisation globale en programmation d.c..

Mémoire présenté pour l'obtention du grade

de licencié en sciences mathématiques

par

DELAUNOIS Anne

TERNET Marie-Pascale

Promoteur:

J.-J. STRODIOT

ANNEE ACADEMIQUE 1987-1988

Facultés Universitaires Notre-Dame de la Paix
Faculté des Sciences
rue de Bruxelles 61,B-5000 **NAMUR**
Tél. 081-22.90.61 Télex 59222 facnam-b Téléfax 081-23.03.91

Etude d'un algorithme de minimisation globale en programmation d.c..

DELAUNOIS Anne
TERNET Marie-Pascale

Résumé
Nous présentons une méthode permettant de calculer le minimum global d'une fonction f sous contraintes $g_i(x) \leq 0$ i=1,...m où f et g_i peuvent s'exprimer comme différence de deux fonctions convexes dans R^n. Après une étude théorique de la méthode de Tuy et un exposé d'une version implémentable, nous proposons diverses modifications en vue d'accélérer la vitesse de convergence de l'algorithme. Nous appliquons ensuite cette méthode à un problème de découpe optimale.

Abstract
A method is presented for computing the global minimum of a function f subject to inequality constraints $g_i(x) \leq 0$ i=1,....m where f and g_i can be expressed as a difference of two convex functions on R^n.First we study a method due to Tuy and we present its implementable version. Then we propose several modifications in order to improve the rate of convergence of the algorithm. Finally this method is applied for solving an optimal shape design problem.

Mémoire de licence en Sciences Mathématiques
Juin 1988
Promoteur : Prof. J.-J. STRODIOT

Que les personnes qui ont contribué de près ou de loin à l'élaboration de ce travail trouvent ici l'expression de notre gratitude.

Nous tenons tout particulièrement à remercier Monsieur J.-J. Strodiot pour ses conseils et sa disponibilité.

Nous adressons nos remerciements les plus vifs aux assistants de l'unité d'optimisation et surtout à Monsieur J. Engels pour l'aide et les remarques judicieuses qu'il nous a fournies tout au long de ce travail.

SOMMAIRE.

CHAPITRE 2. : Recherche des nouveaux sommets et élimination des contraintes redondantes après une coupe.

CHAPITRE 3. : Application à un problème d.c. de découpe de diamants.

CONCLUSION.

INTRODUCTION

Le problème que nous allons aborder consiste à trouver le minimum global d'une fonction f sous un ensemble de contraintes de la forme $x \in \Omega$ et $g_i(x) \leq 0$ où Ω est un ensemble fermé convexe de R^n et f et g_i sont des fonctions non linéaires de R^n à valeurs réelles.

Classiquement pour résoudre de tels problèmes d'optimisation non linéaire, on utilise l'information sur le comportement local des fonctions f et g_i, en cherchant par exemple un point de Kuhn-Tucker . Lorsque le problème est convexe, cette approche conduit à un optimum global, mais en l'absence de convexité, elle ne mène qu'à une solution locale.

Si on ne donne aucune structure aux fonctions f et g_i, il est difficile de traiter le cas général d'une manière déterministe. Heureusement beaucoup de problèmes ont une structure. C'est le cas si on désire minimiser une fonction concave sur un convexe ou maximiser une fonction convexe sur un convexe. Un exemple simple de ce genre de problèmes consiste à chercher, sur un polyèdre connu par ses faces ($Ax \leq b$), le sommet le plus éloigné de l'origine. On voudrait également pouvoir résoudre des problèmes de minimisation d'une fonction s'exprimant comme un maximum de fonctions convexes et concaves ou de maximisation d'une fonction s'exprimant comme un minimum de fonctions convexes et concaves. Un exemple d'un tel problème est le calcul du centre du plus grand cercle inscrit dans un polygône non convexe.

Dans ce mémoire nous désirons montrer que des méthodes efficaces existent pour trouver le minimum ou le maximum global d'un problème de programmation non linéaire lorsque la fonction objectif et les contraintes d'inégalités peuvent s'exprimer comme différence de fonctions convexes (un tel problème s'appelle un problème de programmation d.c.). Nous étudierons plus spécialement la méthode de Tuy [Réf.8]. Elle consiste

à mettre le problème sous une forme dite canonique où la fonction objectif est linéaire et les contraintes sont convexes sauf une qui est anticonvexe. Ensuite des coupes sont effectuées dans un polytope dont on connait les sommets jusqu'au moment où l'optimum global est obtenu. Après avoir démontré la convergence de l'algorithme de Tuy, nous testons une version implémentable de cette méthode [Réf.7] que nous modifions ensuite pour en accélérer la convergence. Deux modifications ont été apportées. La première utilise une méthode due à Horst, de Vries et Thoai [Réf.3] pour trouver les nouveaux sommets du polytope lorsqu'une coupe est effectuée. La deuxième se rapporte à la façon de construire les coupes.

Dans le chapitre 1, après avoir rappelé les propriétés des fonctions d.c. et indiqué comment mettre un programme d.c. sous forme canonique nous présentons une étude de la convergence de l'algorithme de Tuy. Nous indiquons ensuite une version implémentable de celui-ci ainsi que certaines premières modifications. Le chapitre 2 est consacré au problème suivant: étant donnés un polytope dont on connait les sommets et un demi-espace déterminé par un hyperplan,trouver tous les sommets de l'intersection entre ce polytope et ce demi-espace. Enfin dans le chapitre 3 nous appliquons la méthode à un problème de découpe de diamants.

CHAPITRE 1 :

Résolution de problèmes d.c. :

Algorithme de Tuy et versions modifiées.

Avant de décrire l'algorithme de Tuy et les modifications que nous y avons apportées, nous commençons par rappeler les principales propriétés des fonctions d.c.

1.1 Fonctions d.c. et problèmes d.c.

Soit Ω une partie fermée et convexe de R^n. Une fonction $f: \Omega \rightarrow R$ est appelée une fonction d.c. sur Ω si et seulement si elle peut être représentée comme une différence de deux fonctions convexes sur Ω:

$$f(x)=f_1(x)-f_2(x) \qquad (x \in \Omega) \qquad \text{avec } f_1, f_2 \text{ convexes sur } \Omega \ .$$

Toute fonction convexe ou concave sur Ω est évidemment d.c. Pour une étude détaillée des fonctions d.c., nous renvoyons à l'article de Hiriart- Urruty J.-B. [ref. 2]

Citons seulement les résultats suivants:

propriété 1 Toute fonction réelle de classe C^2 sur R^n est d.c. et peut s'écrire comme la différence de deux fonctions convexes, l'une de classe C^2 et l'autre de classe C^∞.

Propriéré 2 La classe des fonctions d.c. sur Ω, DC(Ω), est un espace vectoriel stable pour les opérations $(f_1, f_2) \rightarrow \min \{f_1, f_2\}$

$(f_1, f_2) \rightarrow \max \{f_1, f_2\}$

→ Conv. unif.

Propriété 3 Si Ω est compact, la classe des fonctions d.c. est dense dans C(Ω), c.à.d. toute fonction continue peut être approchée par une fonction d.c. $\|\cdot\|_\infty$ uniformément

La classe des fonctions d.c. "représente" bien les fonctions continues.

Propriété 4 Soit $f_i = g_i - h_i$, où f_i, g_i, h_i sont des fonctions convexes.

On a : a) $$\min_{1\le i\le h} f_i = \sum_{i=1}^{k} g_j - \max_{1\le i\le h} \{ h_j + \sum_{\substack{j=1 \\ j\neq i}}^{k} g_j \}$$

b) $$\max_{1\le i\le h} f_i = -\sum_{i=1}^{k} g_j + \max_{1\le i\le h} \{ g_j + \sum_{\substack{j=1 \\ j\neq i}}^{k} h_j \}$$

Définissons maintenant ce qu'on appelle un problème d.c.

Un problème d.c. est un problème de programmation mathématique de la forme

$$(P1)\begin{cases} \text{minimiser } f(x) \\ \text{s.c. } g_i(x) \ge 0 \quad i=1.....m \\ x \in \Omega \end{cases}$$

où Ω est une partie convexe fermée de R^n et f, $g_1,...,g_m$ sont des fonctions d.c.

Malgré leur grande variété, les programmes d.c.peuvent tous être réduits à une forme canonique que nous allons décrire maintenant. Pour cela, introduisons les définitions suivantes:

1. Une inégalité de la forme $g(x) \geq 0$ où $g: R^n \to R$ est une fonction convexe, est appelée une contrainte anticonvexe .

2. Un problème de programmation mathématique est appelé un programme d.c. mis sous sa forme canonique si sa fonction objectif est linéaire et toutes ses contraintes sont convexes sauf une qui est anticonvexe, c-à-d un programme canonique d.c. est de la forme

$$(P2) \quad \begin{cases} \text{minimiser } c^t x \\ \text{s.c. } x \in \Omega \\ \quad\quad g(x) \geq 0 \end{cases}$$

où $c \in R^n$, Ω est un convexe fermé et $g: R^n \to R$ est une fonction convexe

Proposition 1 Tout programme d.c. peut être converti en un programme canonique d.c.
équivalent.

Démonstration

En ajoutant une variable artificielle v, le programme (P1) s'écrit

$$(P3)\quad \begin{cases} \underset{x,v}{\text{minimiser } v} \\ \text{s.c. } g_i(x) \geq 0 \quad i=1.....m \\ x \in \Omega \\ v-f(x) \geq 0 \end{cases}$$

Notons ensuite $g(x,v) = \min_{1\leq i\leq m} (g_i(x), v-f(x))$.

C'est une fonction d.c. car $g_1,\ldots,g_m$ et v-f(x) sont d.c. et un minimum de fonctions d.c. est d.c. On peut alors écrire g(x,v) sous la forme p(x,v)-q(x,v) avec p,q convexes. (voir propriété 4 pour une façon de calculer effectivement p et q).

Le problème (P3) s'écrit alors

$$(P4)\quad \begin{cases} \underset{x,v}{\text{minimiser } v} \\ \text{s.c. } x \in \Omega \\ p(x,v)-q(x,v) \geq 0 \end{cases}$$

$p(x,v) \geq q(x,v)$
$p(x,v) \geq t \geq q(x,v)$

Si nous considérons une variable supplémentaire t et si nous remarquons que { (x,v)| p(x,v)-q(x,v) ≥ 0 } = { (x,v)| ∃t p(x,v)-t ≥ 0, t-q(x,v) ≥ 0 },alors le problème (P1) devient

$$(P5)\quad \begin{cases} \underset{x,v,t}{\text{minimiser } v} \\ \text{s.c. } x \in \Omega \\ t-q(x,v) \geq 0 \\ p(x,v)-t \geq 0 \end{cases}$$

Comme (P5) est un problème d.c. mis sous forme canonique,on a la thèse.

1.2 Une méthode de résolution de problèmes d.c.

(Tuy 1987)

1.2.1 PRESENTATION GENERALE DE LA METHODE

Nous partirons du problème d.c. convexe mis sous sa forme canonique:

$$(P)\begin{cases}\text{Minimiser } c^t x \\ \text{s.c} \quad h(x) \le 0 \\ \qquad g(x) \le 0\end{cases}$$

où $c \in R^n$, $h : R^n \to R$ convexe et $g : R^n \to R$ concave .

La difficulté du problème (P) est due à la présence de la contrainte concave $g(x) \le 0$. En effet celle-ci supprime la convexité et peut-être même la connexité de l'ensemble des points admissibles.

Remarques

1) Si on a plusieurs contraintes convexes $h_i(x) \le 0$, $i \in I$, on pose

$h(x) = \max \{ h_i(x) \mid i \in I \}$

2) Les fonctions h et g sont continues et sous-différentiables en chaque point (résultat d'analyse convexe [Réf.4]).

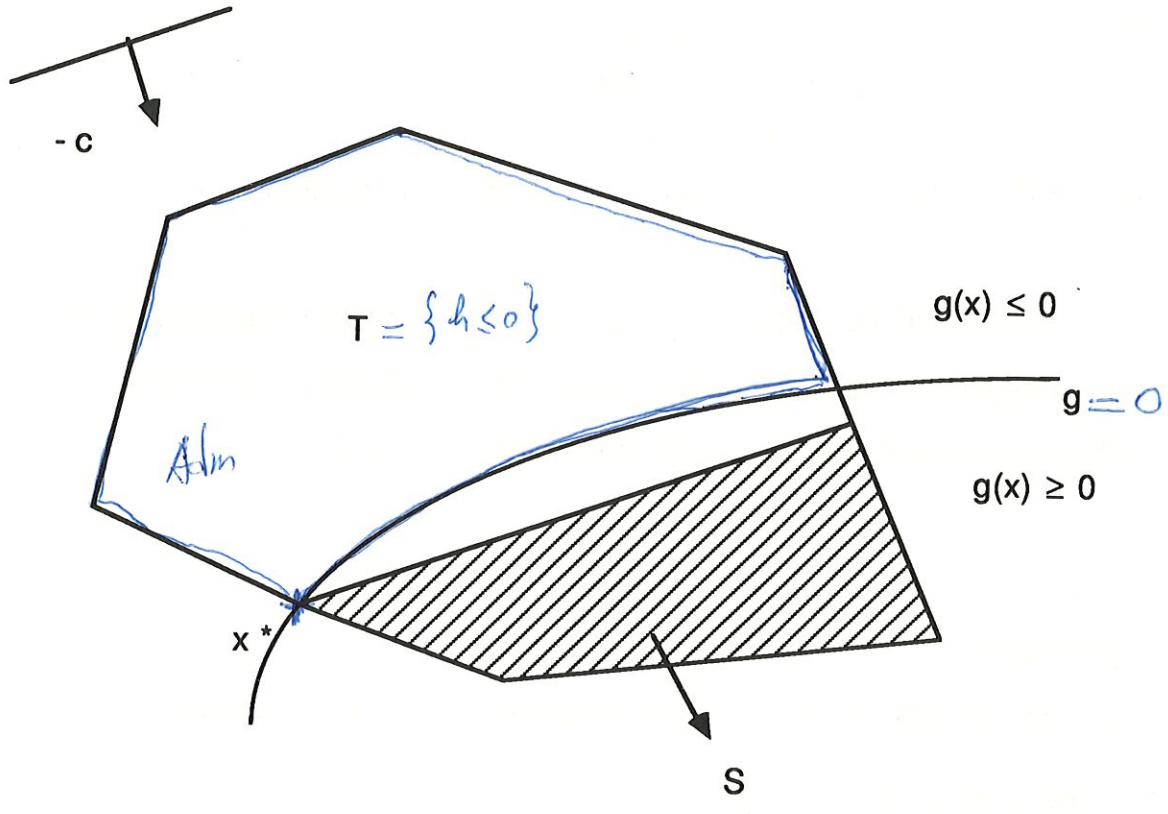


figure 1.

Hypothèses de départ

1) Soit l'ensemble $T = \{ x \in R^n \mid h(x) \leq 0 \}$ fermé et convexe (car h est convexe). Nous supposerons que T est *borné* de sorte que si l'ensemble des points admissibles est non vide , le problème (P) admet une solution .

2) En outre nous allons considérer uniquement le cas où $\min \{ c^t x : h(x) \leq 0 \} < \min \{ c^t x : h(x) \leq 0 \text{ et } g(x) \leq 0 \}$ c'est-à-dire que nous allons supposer que la solution x^* si elle existe, vérifie $g(x^*)=0$. Cette hypothèse est assez naturelle car sinon la contrainte $g(x) \leq 0$ ne jouerait aucun rôle et (P) serait équivalent à un problème convexe ordinaire $\begin{cases} \text{minimiser } c^t x \\ \text{s.c. } h(x) \leq 0 \end{cases}$ pouvant être résolu par de nombreux algorithmes.

3) Nous disposons d'un point ω tel que $\omega \in T$, $g(\omega) > 0$ et $c^t\omega < c^t x^*$ où x^* est solution optimale de (P). (Cette hypothèse étant assez forte, nous verrons par la suite comment s'en débarasser.)

4) Nous disposons également d'un point $\hat{u}^1$ admissible c'est-à-dire tel que $h(\hat{u}^1) \leq 0$ et $g(\hat{u}^1) \leq 0$.

Stratégie

Soit $S = \{ x \in T \mid c^t x \leq c^t x^* \}$ où $c^t x^*$ est la valeur optimale de (P) [fig.1].

Nous cherchons un point x de S tel que $g(x)=0$ (puisque nous avons supposé que la contrainte g était active à l'optimum).

Comme S n'est pas connu, nous allons l'approcher de l'extérieur par une suite décroissante de polytopes (S^k). Nous aurons donc $S^1 \supset S^2 \supset \ldots \supset S^k \supset \ldots \supset S$. Chaque polytope sera obtenu à partir du précédent en effectuant une coupe du type $l(x) \leq 0$ où l est une fonction affine. Plus précisément, $S^{k+1} = S^k \cap \{ x \mid l(x) \leq 0 \}$. Cette coupe sera effectuée de sorte que $S^k \supset S^{k+1}$ et que si $S^k \supset S$ alors $S^{k+1} \supset S$.

Pour approcher la solution x^* de (P), nous essayons à chaque itération de construire un point u^k tel que $u^k \in S^k$ et $g(u^k) = 0$. Dans cette perspective nous allons considérer un point de S^k où g est négative et un autre où g est positive; nous prendrons alors comme u^k l'intersection du segment constitué par ces deux points avec la surface $g(x) = 0$.

Considérons x^k une solution du problème $\begin{cases} \text{minimiser } g(x) \\ \text{s.c. } x \in S^k \end{cases}$

(nous supposons pour le moment que nous pouvons résoudre efficacement ce problème). Comme par hypothèse nous disposons d'un point $\hat{u}^1$ admissible pour (P), nous avons $g(x^k) \leq 0$. Par hypothèse toujours, nous disposons d'un point ω de T tel que $g(\omega) > 0$. Nous pouvons donc calculer un point u^k en prenant l'intersection du segment $[\omega , x^k]$ avec la surface $g(x)=0$.
Ce point u^k peut alors appartenir ou non à T :

1) Si $h(u^k) \leq 0$

Dans ce cas u^k est admissible; nous pouvons éliminer les points x de S^k tels que $c^t x > c^t u^k$. Pour ce faire ,on considère l'hyperplan passant par u^k d'équation $c^t x = c^t u^k$ ainsi que $S^{k+1} = S^k \cap \{ x \mid c^t x \leq c^t u^k \}$. La coupe $c^t x - c^t u^k \leq 0$, notons-la l(x), peut être qualifiée de "bonne coupe

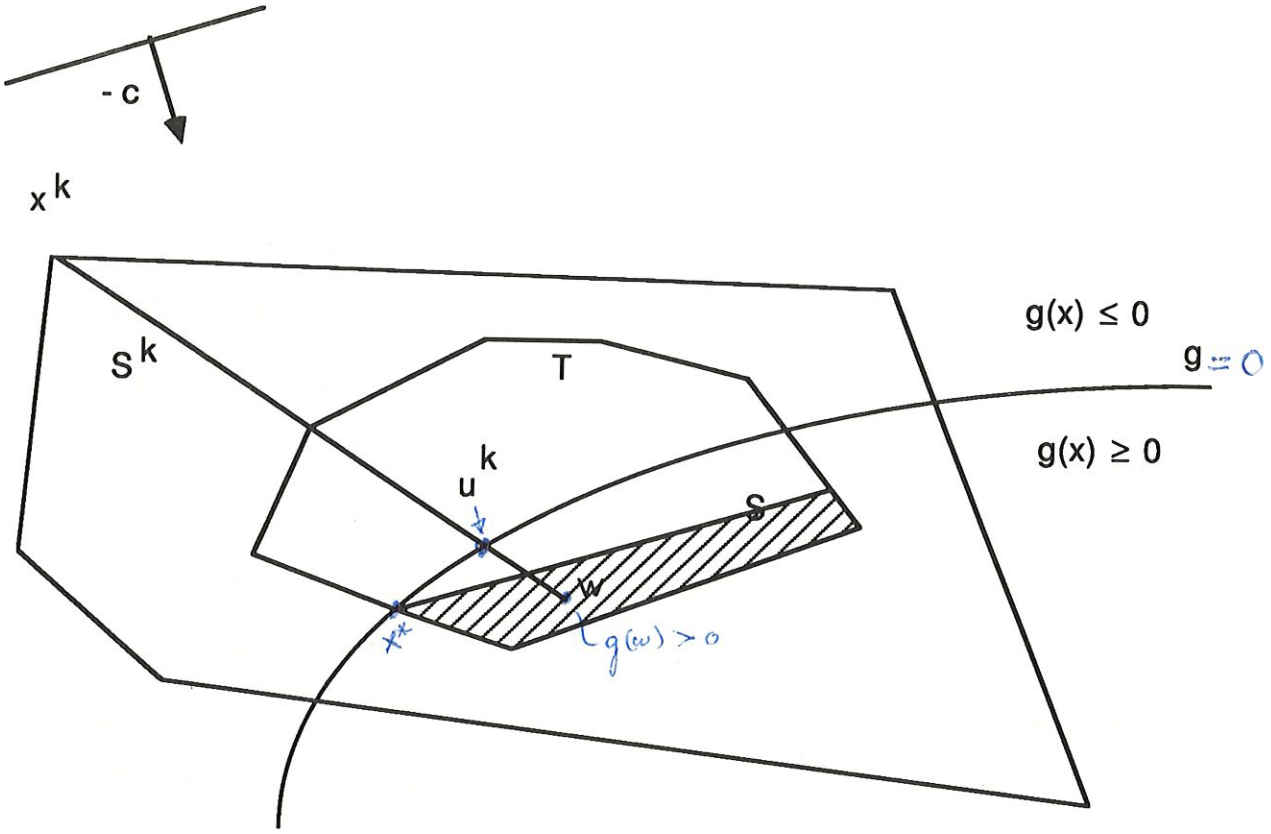


figure 2

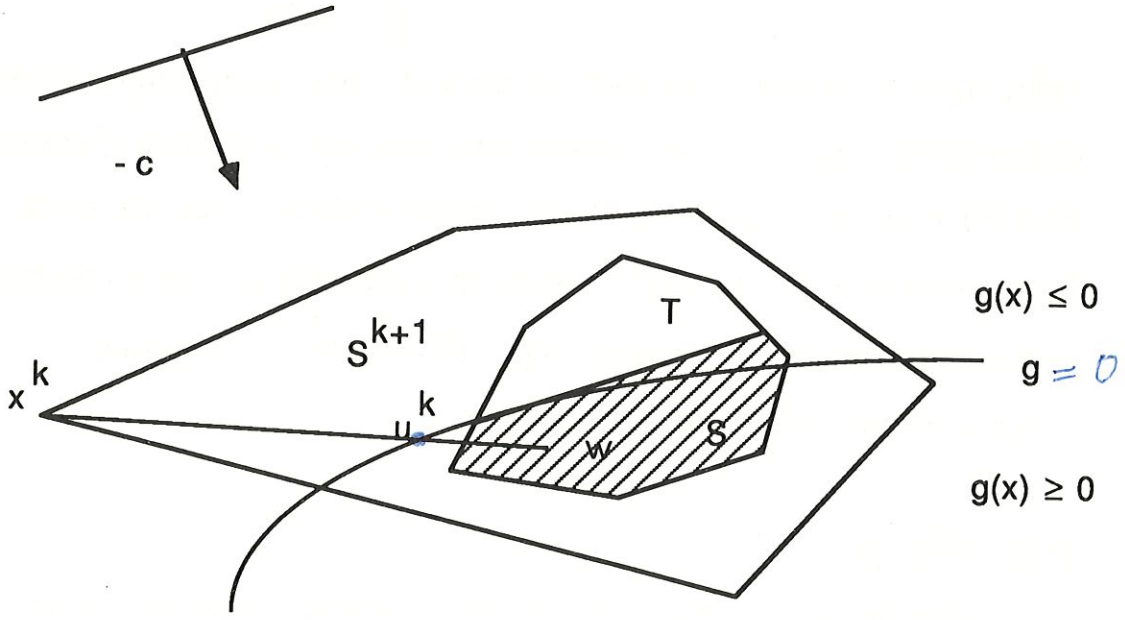


figure 3.

intérieure" puisque, outre le fait qu'elle réduise le polytope S^k, elle supprime une partie de l'ensemble des points admissibles et donc permet de mieux cerner S. [fig 2]. Nous retiendrons la valeur de u^k en posant $\hat{u}^{k+1} = u^k$.

Le polytope S^{k+1} résultant de la coupe contient toujours S. En effet, on a $S^{k+1} \supset \{ x \in T \mid c^t x \leq c^t u^k \} \supset \{ x \in T \mid c^t x \leq c^t x^* \} = S$
La première inclusion est vraie par construction des coupes et de S^1, la deuxième car u^k est admissible pour (P).

→ **2) Si $h(u^k) > 0$** c.à.d $u^k \notin T$

Dans ce deuxième cas,nous effectuons une coupe en vue de réduire S^k mais cette fois nous ne supprimons aucun point de T; la coupe est extérieure à T.

Notons par $\partial h(u^k)$ le sous-différentiel de h au point u^k.
Soit $t^k \in \partial h(u^k)$. Nous effectuons la coupe $l(x) = (x-u^k)t^k + h(u^k) \leq 0$ et nous définissons $S^{k+1} = S^k \cap \{ x \mid l(x) \leq 0 \}$. [fig. 3] Nous poserons alors $\hat{u}^{k+1} = \hat{u}^k$.

Le polytope S^{k+1} contient toujours S. En effet, soit x arbitraire dans S. Comme h est convexe, la définition du sous-différentiel implique que $l(x) = (x-u^k)t^k + h(u^k) \leq h(x) \leq 0$. D'où $x \in S^{k+1}$ et donc $S^{k+1} \supset S$.

Il reste à voir comment résoudre le problème $\begin{cases} \text{minimiser } g(x) . \\ \text{s.c. } x \in S^k \end{cases}$

Or, comme on minimise une fonction concave sur un convexe S^k,on sait

que le minimum global est atteint en un sommet de S^k. Ainsi si nous notons par $V(S^k)$ l'ensemble des sommets du polytope S^k, on obtient que x^k est solution de $\begin{cases} \text{minimiser } g(x) \\ \text{s.c. } x \in V(S^k) \end{cases}$,ce qui est très simple à résoudre.

1.2.2 ALGORITHME

Notons V(S) l'ensemble des sommets d'un polytope S, $\partial h(x)$ le sous-différentiel de h au point x et argmin{ g(x) | $x \in S$ } l'ensemble des solutions du problème $\begin{cases} \text{minimiser } g(x) \\ \text{s.c. } x \in S \end{cases}$. L'algorithme correspondant à la méthode décrite plus haut s'écrit :

Algorithme

PAS 1 : Choisir un point ω tel que $h(\omega) < 0$, $g(\omega) > 0$ et $c^t\omega < c^tx^*$

Choisir un point $\hat{u}^1$ admissible pour (P)

Engendrer un polytope S^1 contenant $T=\{ x \in R^n \mid h(x) \leq 0 \}$

Calculer $V(S^1)$ et poser k=1

PAS 2 : Calculer $x^k \in$ argmin { g(x) | $x \in V(S^k)$ }

2.a. Si $g(x^k)=0$ alors $\hat{u}^k$ est solution de (P) [cfr. Thm 1 en 1.2.3]

2.b. Sinon ($g(x^k)<0$) calculer le point $u^k = [\omega, x^k] \cap \{ x \mid g(x)=0 \}$ et aller au pas 3.

PAS 3 :

3.a. Si $h(u^k) \leq 0$

poser $\hat{u}^{k+1} = u^k$, $l(x)=c^tx-c^tu^k$ et aller au pas 4.

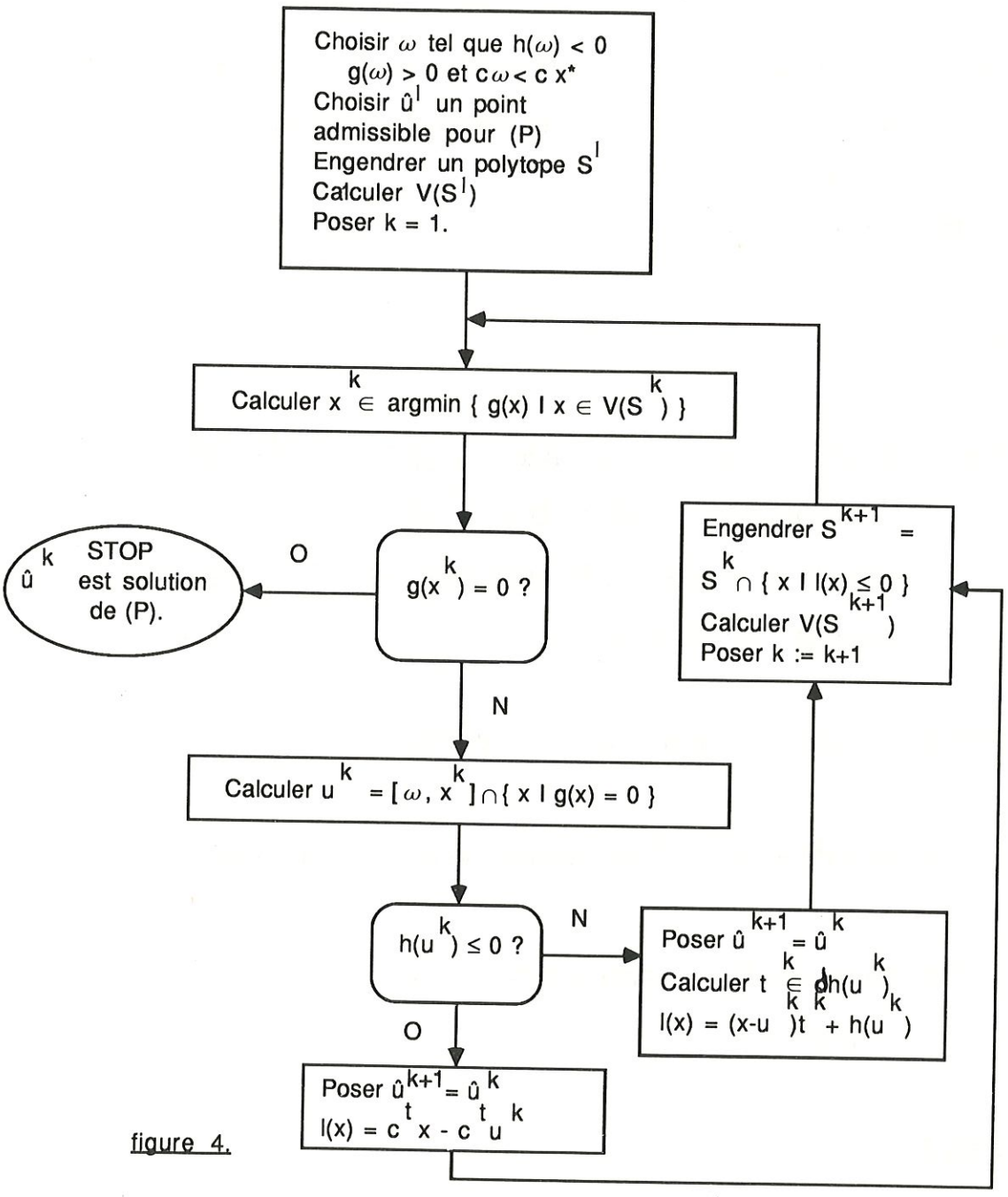

Choisir ω tel que h(ω) < 0
g(ω) > 0 et cω < c x*
Choisir û^l un point admissible pour (P)
Engendrer un polytope S^l
Calculer V(S^l)
Poser k = 1.
Calculer x^k ∈ argmin { g(x) I x ∈ V(S^k) }
g(x^k) = 0 ?
O
STOP
û^k est solution de (P).
N
Calculer u^k = [ω, x^k] ∩ { x I g(x) = 0 }
h(u^k) ≤ 0 ?
N
O
Poser û^k+1 = û^k
l(x) = c^t x - c^t u^k
Engendrer S^k+1 = S^k ∩ { x I l(x) ≤ 0 }
Calculer V(S^k+1)
Poser k := k+1
Poser û^k+1 = û^k
Calculer t^k ∈ ∂h(u^k)
l(x) = (x-u^k)t^k + h(u^k)

figure 4.

3.b. Sinon ($h(u^k)>0$)

poser $\hat{u}^{k+1}=\hat{u}^k$, $l(x)=(x-u^k)\, t^k + h(u^k)$ et aller au pas 4.

PAS 4 Engendrer un nouveau polytope $S^{k+1}= S^k \cap \{ x \mid l(x) \leq 0 \}$

calculer $V(S^{k+1})$, poser $k=k+1$ et retourner au pas 2.

[fig. 4]

1.2.3. CONVERGENCE DE LA METHODE

Nous ferons l'hypothèse supplémentaire que le problème (P) est ***stable*** c'est-à-dire que

$$\min \{ c^t x \mid h(x) \leq 0 , g(x) \leq -\varepsilon \} \downarrow \min \{ c^t x \mid h(x) \leq 0 , g(x) \leq 0 \}$$

quand $\varepsilon \downarrow 0$.

Le lemme suivant donne une condition nécessaire et suffisante pour qu'un point x soit solution du problème (P).

Lemme 1

Si le problème (P) est stable et si $\bar{x}$ est un point admissible de (P)
Alors $\bar{x}$ est un minimum global de (P) si et seulement si

$$0 = \min \{ g(x) \mid x \in T , c^t x \leq c^t \bar{x} \}$$

Démonstration

Comme $\bar{x}$ est admissible pour (P) nous avons que

$$\min \{ g(x) \mid x \in T , c^t x \leq c^t \bar{x} \} \leq 0$$

$\Rightarrow$ Si nous supposons par l'absurde que $0 > \min \{ g(x) \mid x \in T , c^t x \leq c^t \bar{x} \}$, alors $\exists\, x \in T$ tel que $g(x) < 0$ et $c^t x \leq c^t \bar{x}$; ce qui contredit le fait que $\bar{x}$ soit minimum global.

$\Leftarrow$ Par hypothèse il n'existe pas de x dans T tel que $c^t x \leq c^t \bar{x}$ et $g(x) < 0$.

Soit $\varepsilon > 0$ arbitraire. Notons x^ε une solution optimale du problème

perturbé $\begin{cases} \text{minimiser } c^t x \\ \text{s.c. } x \in T \\ \qquad g(x) \leq -\varepsilon \end{cases}$

Comme $g(x^\varepsilon) < 0$ et $x^\varepsilon \in T$ nous devons avoir $c^t x^\varepsilon > c^t \bar{x}$.

Si $\varepsilon \downarrow 0$ nous avons $c^t x^\varepsilon \rightarrow c^*$ car (P) est stable. D'où $c^* \geq c^t \bar{x}$.

Comme $c^* \leq c^t x \ \forall$ x admissible, nous avons la thèse.

Proposition 1

Supposons que le problème (P) soit stable.
Si $g(x^k)=0$ alors le point $\hat{u}^k$ est solution de (P).

Démonstration

Comme $\hat{u}^k$ est admissible pour (P), en vertu du lemme 1 il suffit de montrer que $\min \{ g(x) \mid x \in T , c^t x \leq c^t \hat{u}^k \} \geq 0$.

Or puisque $S^k \supset \{ x \mid x \in T , c^t x \leq c^t \hat{u}^k \}$, nous avons en utilisant la définition de x^k que

$$0 = g(x^k) = \min \{ g(x) \mid x \in S^k \} \leq \min \{ g(x) \mid x \in T , c^t x \leq c^t \hat{u}^k \}$$

et donc nous avons la thèse.

Propositon 2

Supposons que le problème (P) soit stable.
Si la suite (x^k) a un point d'accumulation $\bar{x}$ tel que $g(\bar{x}) = 0$, alors tout point d'accumulation de la suite $(\hat{u}^k)$ est une solution optimale de (P).

Démonstration

Soit $\hat{u}$ un point d'accumulation de la suite $(\hat{u}^k)$.
En vertu du lemme 1, il suffit de montrer que $\hat{u}$ est admissible pour (P) et que $0 = \min \{ g(x) \mid x \in T , c^t x \leq c^t \hat{u} \}$.

1) $\hat{u}$ est admissible pour (P) :

Comme g et h sont continues et comme, par construction, $g(\hat{u}^k) \leq 0$ et $h(\hat{u}^k) \leq 0$ nous avons immédiatement que $g(\hat{u}) \leq 0$ et $h(\hat{u}) \leq 0$, c'est-à-dire que $\hat{u}$ est admissible.

2) $\min \{ g(x) \mid x \in T , c^t x \leq c^t \hat{u} \} = 0$:

Comme $\hat{u} \in T$ et $g(\hat{u}) \leq 0$ nous avons immédiatement que le membre de gauche est ≤ 0. Montrons l'inégalité inverse.

Par définiton de x^k nous avons que $g(x^k) \leq g(x) \ \forall \ x \in S^k$ et donc que

$$g(x^k) \leq g(x) \ \ \forall \ x \in \bigcap_{k=1}^{\infty} S^k \ \ \forall \ k.$$

Comme g est continue et comme $g(\bar{x}) = 0$, on en déduit

$$0 = g(\bar{x}) \leq \min \{ g(x) \mid x \in \bigcap_{k=1}^{\infty} S^k \} \quad (1)$$

D'autre part, par construction, nous avons que $(c^t\hat{u}^k)$ est une suite décroissante à partir d'un certain rang et que pour tout k

$$S^k \supset \{ x \mid h(x) \leq 0 \text{ et } c^t x \leq c^t\hat{u}^k \}$$

D'où

$$\bigcap_{k=1}^{\infty} S^k \supset \{ x \mid h(x) \leq 0 \text{ et } c^t x \leq c^t\hat{u}^k \}$$

et donc en vertu de **(1)**

$$0 \leq \min \{ g(x) \mid h(x) \leq 0 \text{ et } c^t x \leq c^t\hat{u}^k \}$$

La thèse est alors immédiate.

Lemme 2

Soit D un ensemble arbitraire.

Soit (x^k) une suite bornée de R^n.

Supposons que pour k=1,2,.... il existe une fonction affine $l^k(.)$ telle que

a) $l^k(v) \leq 0$ pour un v fixé; $l^k(x^k) > 0$

b) $l^j(x^k) \leq 0 \quad \forall j < k$

c) pour toute sous-suite (x^{k_n}) telle que $x^{k_n} \to \bar{x} \notin D$ et $l^{k_n} \to l(x) \quad \forall x \in R^n$, on a $l(\bar{x}) > 0$

Alors tout point d'accumulation de x^k appartient à D.

Pour la démonstration voir [Réf.10]

Théorème 1

Supposons que le problème (P) soit stable .
Si l'algorithme est infini, alors tout point d'accumulation $\hat{u}$ de la suite $(\hat{u}^k)$ est une solution optimale de (P).

Démonstration

En vertu de la proposition 2, il suffit de montrer que tout point d'accumulation $\bar{x}$ de la suite (x^k) vérifie $g(\bar{x})=0$.
Pour cela vérifions que les conditions du lemme 2 sont satisfaites pour $D=\{ x \mid g(x) = 0\}$ et $v = \omega$.

a) $l^k(\omega) \leq 0$ et $l^k(x^k) > 0$

- Si $l^k(x)$ est de la forme $l^k(x) = c^t x - c^t u^k$, alors $l^k(\omega) = c^t\omega - c^t u^k < 0$ par définition de ω. De plus comme $l^k(u^k) = 0$ et $u^k = \lambda x^k + (1-\lambda)\,\omega$ avec $0 < \lambda < 1$ (car l'algorithme est infini), nous avons $l^k(x^k) > 0$.

- Si par contre $l^k(x)$ est de la forme $l^k(x) = (x-u^k)t^k + h(u^k)$, alors $l^k(\omega) = (\omega-u^k)t^k + h(u^k) \leq h(\omega) < 0$ par définition de ω et du sous-gradient t^k. De plus par construction de l'algorithme nous avons que $l^k(u^k) = h(u^k) > 0$ et donc $l^k(x^k) > 0$ car $u^k = \lambda x^k + (1-\lambda)\,\omega$.

b) $l^j(x^k) \leq 0 \quad \forall j < k$

Cette condition est évidente par construction de l'algorithme.

c) Pour toute sous-suite x^{k_n} telle que $x^{k_n} \to \bar{x} \notin D$ et $l^{k_n}(x) \to l(x)$ $\forall x \in R^n$ on a $l(\bar{x}) > 0$.

Comme $g(\bar{x}) \neq 0$ et $g(x^k) < 0$ $\forall k$, nous avons évidemment que $g(x) < 0$.

En prenant *une sous-suite si nécessaire* nous pouvons considérer que l'un des deux cas suivants a toujours lieu :

1) $u^{k_n} \in T$ $\forall n$

Dans ce cas $l^{k_n}(x)$ est de la forme $c^t x - c^t u^{k_n}$. En prenant une sous-suite si nécessaire, nous pouvons supposer que $u^{k_n} \to \bar{u}$. Par définition de ω $l(\omega) < 0$. D'autre part comme $g(x) < 0$, nous avons $x = \bar{u} + \lambda(\bar{u}-\omega)$ avec $\lambda > 0$ et donc $l(\bar{x}) > 0$.

2) $u^{k_n} \notin T$ $\forall n$

Dans ce cas $l^{k_n}(x)$ est de la forme $(x-u^{k_n})\, t^{k_n} + h(u^{k_n})$.

Comme $t^{k_n} \in \delta h(u^{k_n})$ et que (u^{k_n}) est bornée nous avons que (t^{k_n}) est bornée [Réf.4]. D'où en prenant une sous-suite si nécessaire, nous pouvons supposer que $u^{k_n} \to \bar{u}$ et $t^{k_n} \to t$ avec $t \in \delta h(\bar{u})$. Par définition de ω et du sous-gradient t, nous avons que

$$l(\omega) = (\omega-\bar{u})t + h(\bar{u}) \leq h(\omega) < 0$$

et par hypothèse

$$l(\bar{u}) = h(\bar{u}) > 0.$$

Comme $g(\bar{x}) < 0$, nous avons $\bar{x} = \bar{u} + \lambda(\bar{u}-\omega)$ avec $\lambda > 0$ et donc $l(\bar{x}) > 0$.

Les conditions du lemme 2 sont donc vérifiées pour $D=\{ x \mid g(x) = 0\}$ et $v = \omega$.

1.3 Un algorithme implémentable (N.V. Thoai)

1.3.1 DESCRIPTION

Dans l'algorithme décrit au paragraphe précédent,nous supposions disposer d'un point ω de T tel que $g(\omega) > 0$ et $c^t\omega < c^tx^*$ (x^* est solution optimale du problème (P)). Pour obtenir un tel point il faudrait minimiser la fonction objectif c^tx sur les points de T c'est-à-dire résoudre le problème convexe $\begin{cases}\text{minimiser } c^tx \\ \text{s.c. } x \in T\end{cases}$. Dans une version implémentable, nous ne désirons pas résoudre un tel problème avant de débuter l'algorithme.

A chaque itération, nous allons plutôt considérer le point z^k, solution du problème $\begin{cases}\text{minimiser } c^tx \\ \text{s.c. } x \in S^k\end{cases}$ qui nous donne une borne inférieure sur la valeur optimale de (P). Comme nous ne connaissons pas a priori le signe de la fonction g au point z^k, nous ne pouvons garantir de trouver un point u^k sur la surface g à chaque itération. Dès lors, deux cas sont à envisager:

1) $g(z^k) > 0$

Comme $g(x^k) \leq 0$ (sinon le problème (P) n'a pas de solution), nous remplaçons le point ω par le point z^k et u^k est défini comme l'intersection du segment $[x^k,z^k]$ avec la surface $g(x)=0$.

Si $h(u^k) \leq 0$ alors u^k est admissible et lorsque $c^tu^k = c^tz^k$, u^k est solution de (P) car c^tz^k est une borne inférieure sur la valeur optimale de

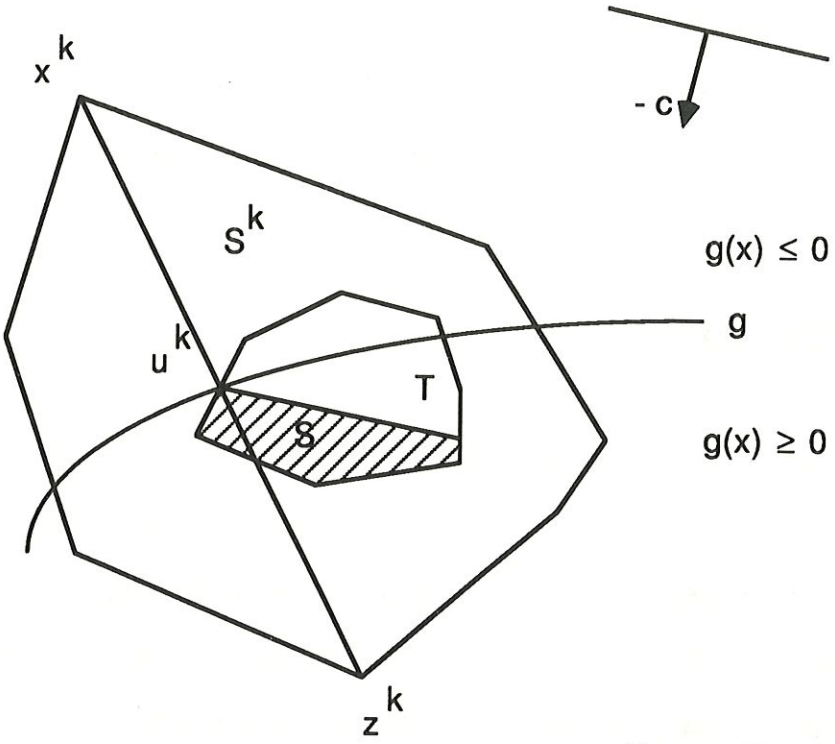


figure 5

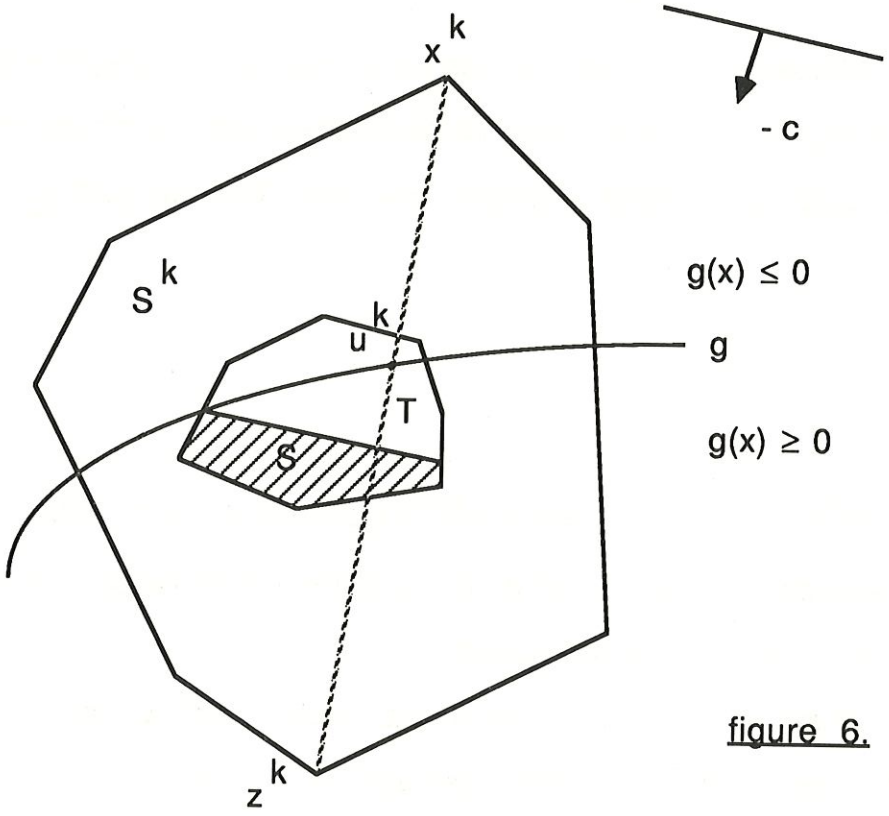


figure 6.

(P) . [fig.5] Lorsque $c^t u^k \neq c^t z^k$, nous effectuons une coupe du type $l(x) = c^t u^k - c^t z^k \leq 0$. [fig 6]

Si $h(u^k) > 0$ alors u^k n'est pas solution de (P) et la coupe effectuée est du type $l(x) = (x-u^k)t^k + h(u^k) \leq 0$. [fig.7]

2) $g(z^k) \leq 0$

Nous ne pouvons plus construire par le procédé précédent de point u^k vérifiant $g(u^k)=0$. Dès lors, il est aussi inutile de calculer x^k.

Si $h(z^k) \leq 0$ alors z^k est solution optimale de (P). [fig. 8]

Sinon, nous effectuons une coupe du type $l(x) = (x-z^k)t^k + h(z^k) \leq 0$. [fig. 9]

Remarques

1) Le problème $\begin{cases} \text{minimiser } c^t x \\ \text{s.c. } x \in S^k \end{cases}$ permettant de trouver z^k se ramène au

problème $\begin{cases} \text{minimiser } c^t x \\ \text{s.c. } x \in V(S^k) \end{cases}$ puisqu'on minimise une fonction linéaire sur un convexe S^k.

2) Dans le cas où on dispose d'un point $b \in T$, les coupes effectuées à partir

d'un point extérieur à T peuvent être faites de la façon suivante :

si e désigne ce point extérieur alors on calcule l'intersection du segment [b,e] avec la surface $h(x)=0$. Si nous notons y^k ce point d'intersection , la coupe est alors donnée par

$$(x-y^k)t^k + h(y^k) \leq 0 \qquad (t^k \in \delta h(y^k))$$

[fig. 10 et fig. 11]

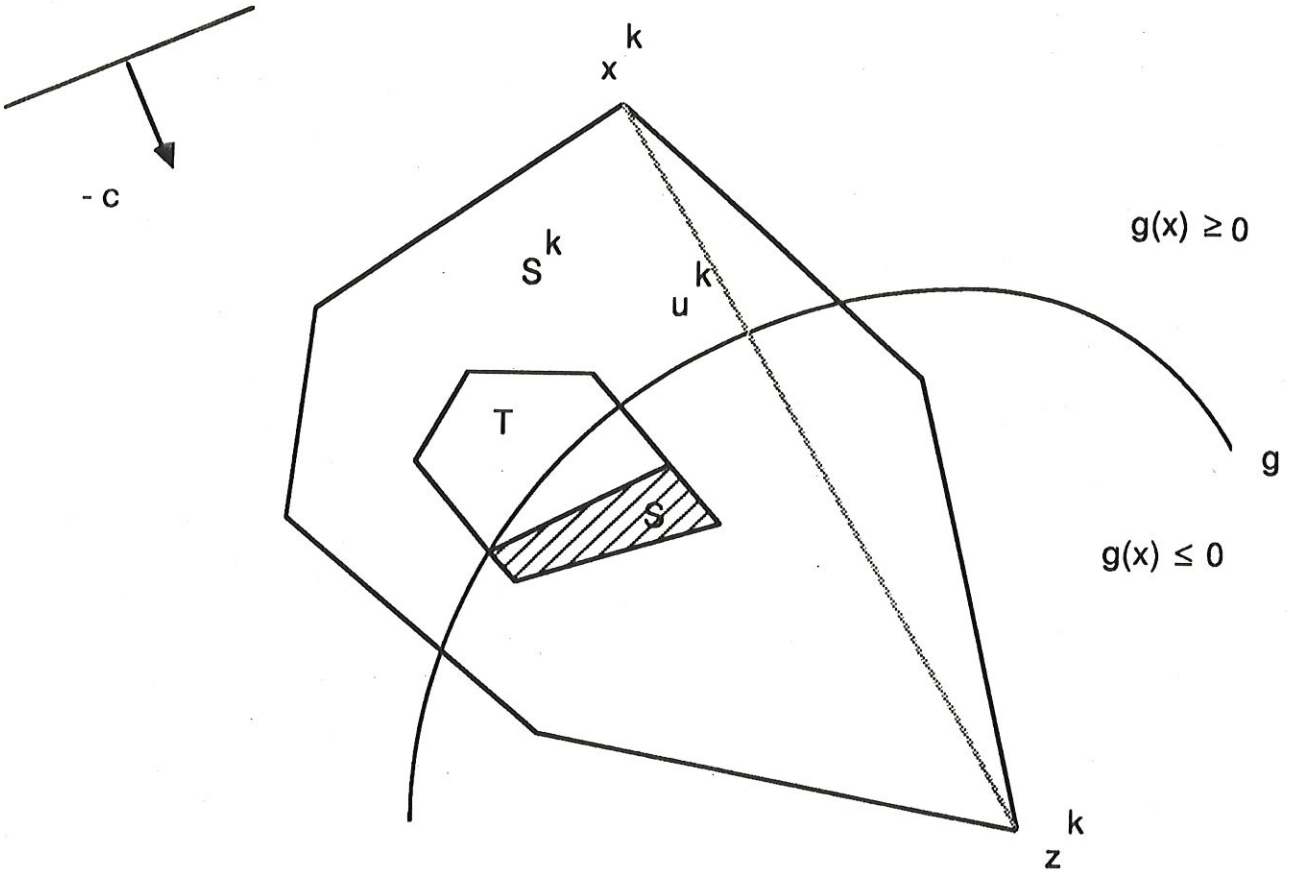


figure 7

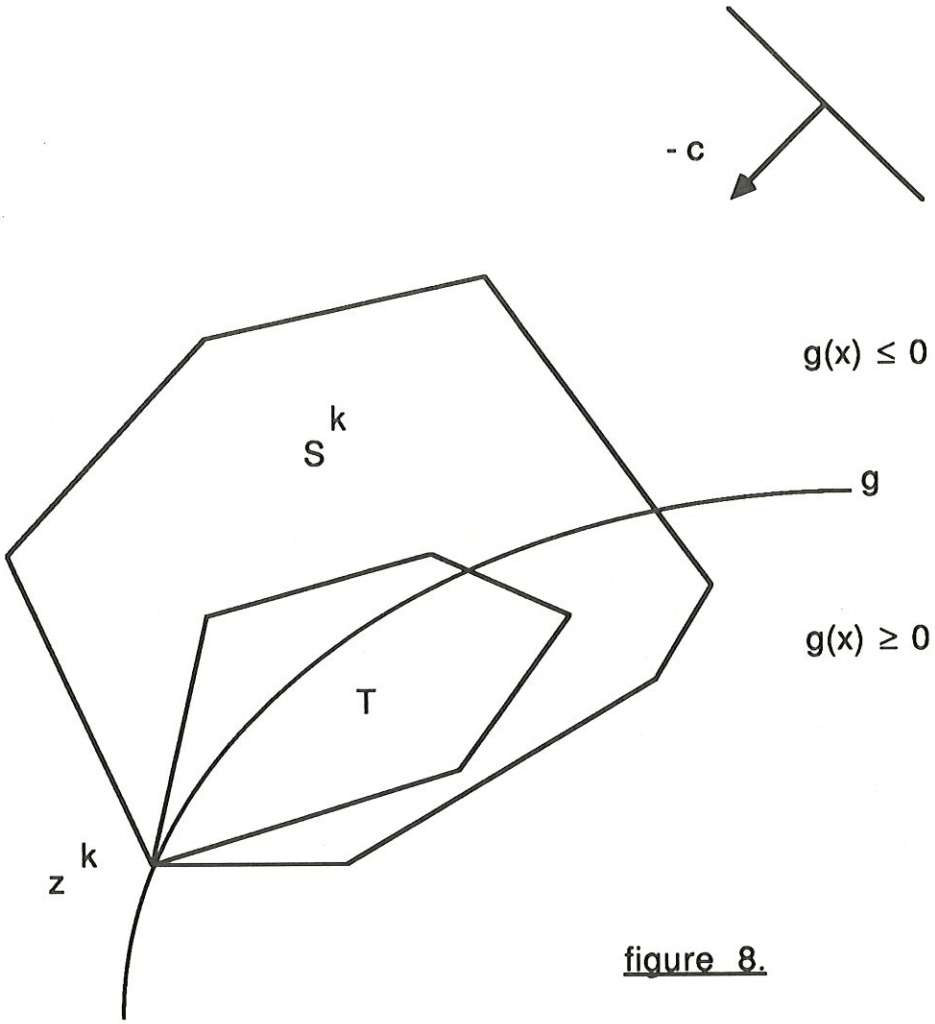


figure 8.

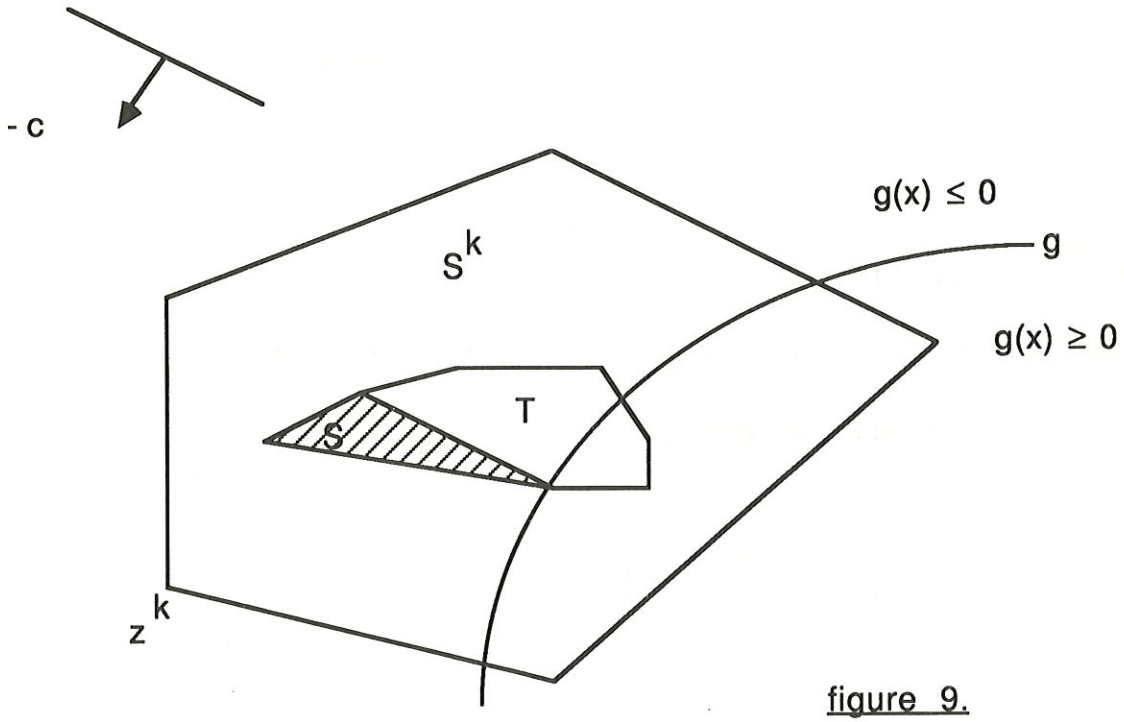


figure 9.

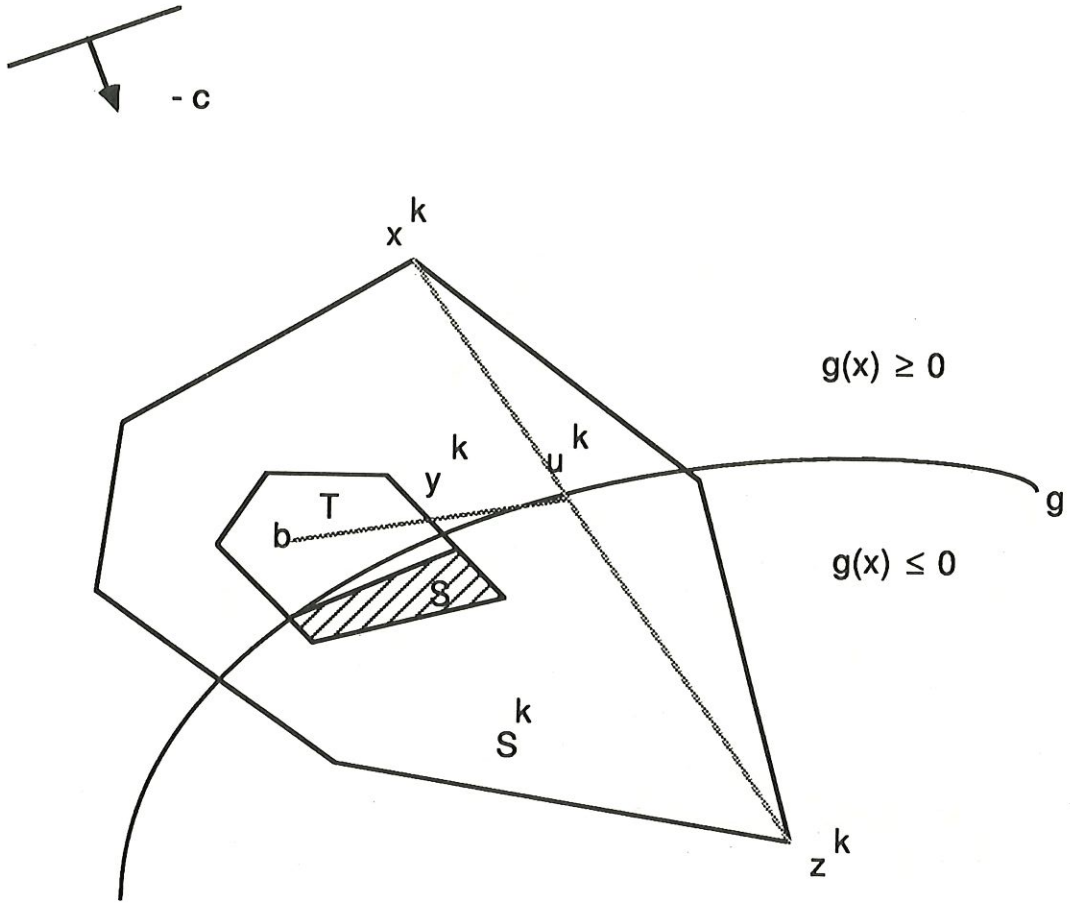
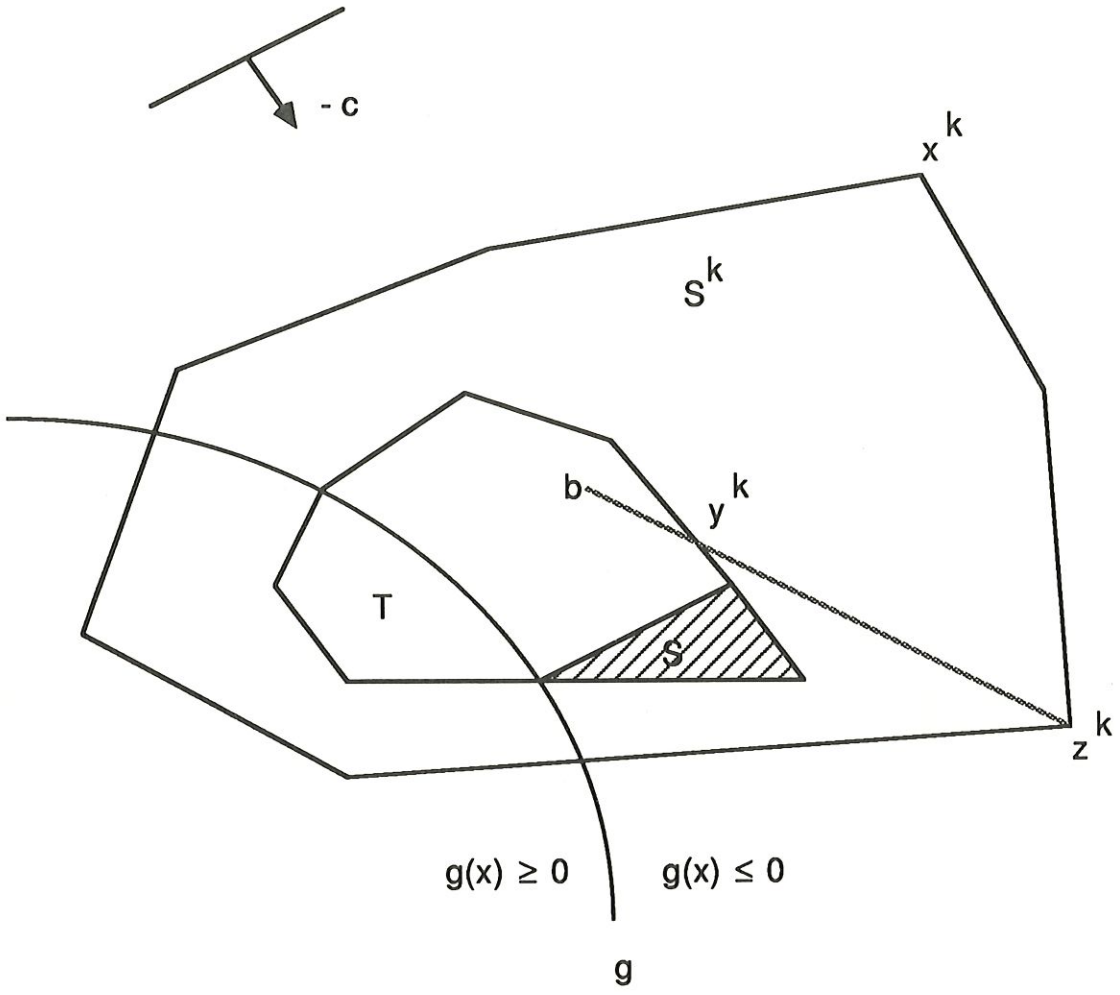


figure 10

$g(z^k) > 0$ et $h(u^k) > 0$

Dans ce cas on calcule $y^k = [\, b\, ,\, u^k\,] \cap \{ x : h(x) \leq 0 \}$ et on effectue une coupe du type :

$$l(x) = (x - y^k)\, t^k + h(y^k) \leq 0$$



$g(z^k) \leq 0$ et $h(z^k) > 0$

On cacule $y^k = [\, b\, ,\, z^k\,] \cap \{\, x : h(x) = 0 \,\}$ et on effectue la coupe $l(x) = (x-y^k)\, t^k + h(y^k) \leq 0$

figure 11.

1.3.2 ALGORITHME MODIFIE

PAS 1 : Engendrer un polytope S^1 contenant $T=\{ x \in R^n \mid h(x) \leq 0 \}$.

Calculer $V(S^1)$ et poser $k=1$.

PAS 2 : Calculer $z^k \in$ argmin $\{ c^t x \mid x \in V(S^k) \}$

2.a. Si $h(z^k) \leq 0$ (càd $z^k \in T$) alors

Si $g(z^k) \leq 0$ alors STOP : z^k est solution

2.b. Si $h(z^k) > 0$ (càd $z^k \notin T$) alors

Si $g(z^k) \leq 0$ alors poser $y^k=z^k$ et aller au pas 4

Sinon, aller au pas 3.

PAS 3 : Calculer $x^k \in$ argmin $\{ g(x) \mid x \in V(S^k) \}$

3.a. Si $g(x^k) > 0$ alors STOP : pas de solution admissible.

3.b. Si $g(x^k) = 0$ et qu'il existe un point v calculé à une itération précédente $k' < k$ par 3c(1), alors STOP : v est solution.

3.c. Sinon ($g(x^k) < 0$)

calculer le point $u^k = [x^k, z^k] \cap \{ x \mid g(x)=0 \}$ (Remarquons que u^k unique parce que $\{ x \mid g(x)>0 \}$ est convexe).

Si $h(u^k) > 0$ alors poser $y^k=u^k$ et aller au pas 4

Si $h(u^k) \leq 0$ et $c^t u^k = c^t z^k$ alors STOP : u^k est solution

Sinon ($h(u^k) \leq 0$ et $c^t u^k \neq c^t z^k$) poser $v=u^k$ et aller au pas 5. (1)

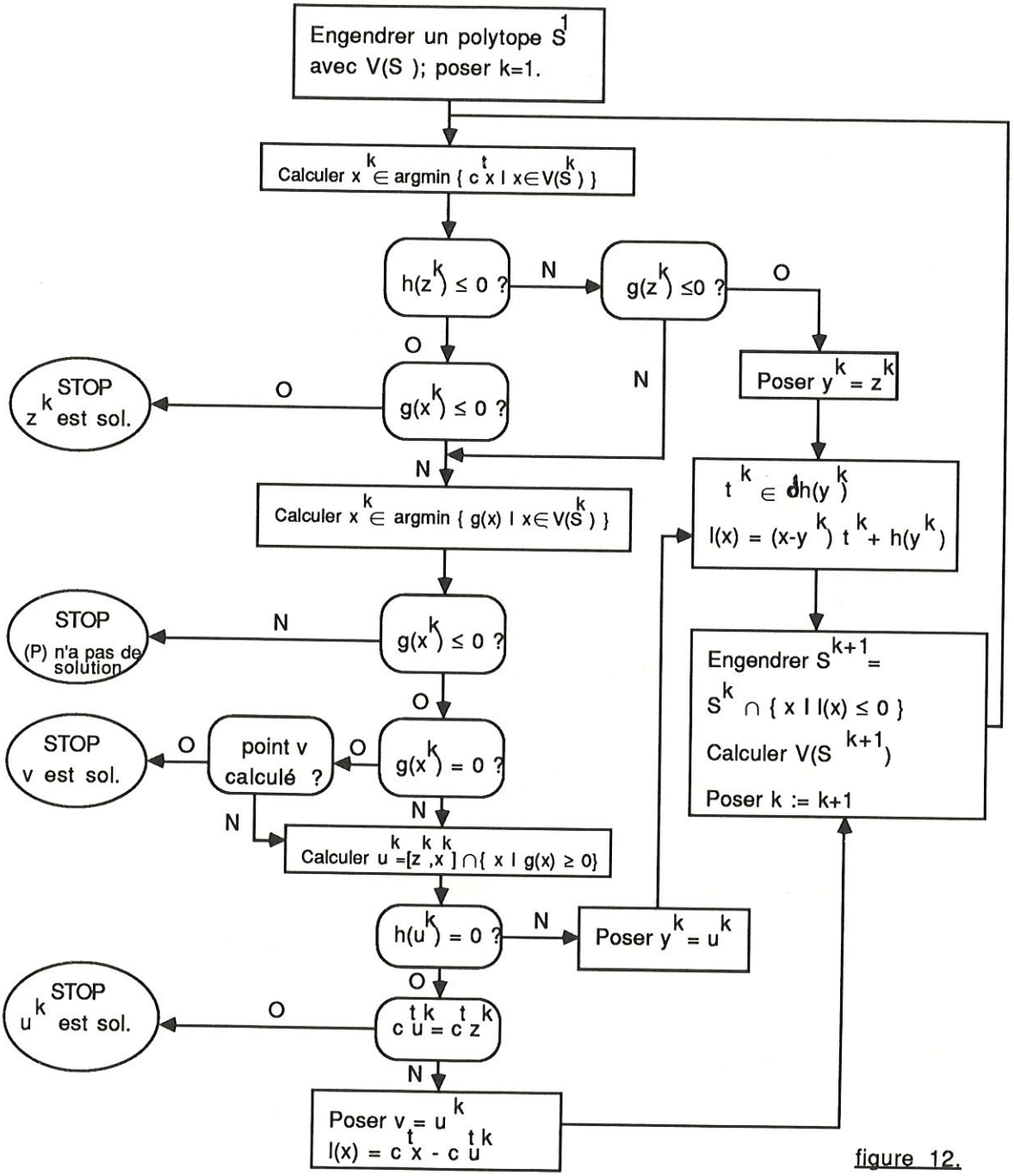

Engendrer un polytope S^1 avec V(S); poser k=1.
Calculer $x^k \in \text{argmin} \{ c^t x \mid x \in V(S^k) \}$
$h(z^k) \le 0$?
N
$g(z^k) \le 0$?
O
Poser $y^k = z^k$
O
STOP z^k est sol.
O
$g(x^k) \le 0$?
N
N
Calculer $x^k \in \text{argmin} \{ g(x) \mid x \in V(S^k) \}$
$t^k \in \partial h(y^k)$
$l(x) = (x-y^k)\, t^k + h(y^k)$
STOP (P) n'a pas de solution
N
$g(x^k) \le 0$?
O
Engendrer $S^{k+1} = S^k \cap \{ x \mid l(x) \le 0 \}$
Calculer $V(S^{k+1})$
Poser k := k+1
STOP v est sol.
O
point v calculé ?
O
$g(x^k) = 0$?
N
N
Calculer $u^k = [z^k, x^k] \cap \{ x \mid g(x) \ge 0 \}$
$h(u^k) = 0$?
N
Poser $y^k = u^k$
O
STOP u^k est sol.
O
$c^t u^k = c^t z^k$
N
Poser $v = u^k$
$l(x) = c^t x - c^t u^k$

figure 12.

PAS 4 : Calculer le vecteur $t^k \in \delta h(y^k)$ et

le plan de coupe $l(x)=(x-y^k)t^k+h(y^k)$.

Aller au pas 5.

PAS 5 : Engendrer un nouveau polytope $S^{k+1} = S^k \cap \{ x \mid l(x) \leq 0 \}$ et

calculer $V(S^{k+1})$.

Poser $k=k+1$ et retourner au pas 2.

[fig. 12]

1.3.3 REMARQUES

Construction d'un polytope S^1 (au pas 1)

Pour initialiser l'algorithme, nous avons besoin d'un polytope S^1 contenant $S=\{ x \in T \mid c^t x \leq c^t x^* \}$ et de l'ensemble des sommets de S^1, $V(S^1)$. Pour être sûr que S^1 contienne bien S, nous allons le construire de sorte qu'il contienne l'ensemble convexe compact $T= \{ x \in R^n \mid h(x) \leq 0 \}$. Ce polytope sera décrit par un système d'inéquations . Nous allons essayer de le construire avec la structure la plus simple possible, c'est-à-dire de sorte que le nombre d'inéquations qui détermine le polytope et donc le nombre de ses sommets soit le plus petit possible. Un tel polytope est un n-simplexe dans R^n.

Si T se trouve dans R^n_+ , nous pouvons choisir simplement

$$S^1 = \{ x \in R^n \mid x_i \geq 0 \;\; i=1,...n \,, \sum_{i=1}^{n} x_i \leq \gamma \}$$

où γ est un nombre suffisamment grand tel que

$$\gamma \geq \max \{ \sum_{i=1}^{n} x_i \mid x \in T \}$$

Le n-simplexe S^1 possède n+1 sommets, à savoir l'origine et n points où l'hyperplan $\{ x \mid \sum_{i=1}^{n} x_i = \gamma \}$ rencontre les axes.

Les n+1 sommets de S^1 sont

$$(0,0,0,.....,0)$$

$$(\gamma,0,0,.....,0)$$

$(0, \gamma, 0, \ldots\ldots\ldots\ldots\ldots\ldots, 0)$

$(0, 0, \gamma, \ldots\ldots\ldots\ldots\ldots\ldots, 0)$

...

$(0, 0, 0, \ldots\ldots\ldots\ldots\ldots\ldots, \gamma)$

Dans le cas général, un n-simplexe peut être choisi comme suit :

$$S^1 = \{ x \in R^n \mid x_i \geq -\gamma \; i=1,\ldots n \quad \sum_{i=1}^{n} x_i \leq \gamma \}$$

où γ est un nombre suffisamment grand tel que

$$\gamma \geq \max \{ \mid \min \{x_i \mid x \in T \; i=1,\ldots n\} \mid \; ; \mid \max \{ \sum_{i=1}^{n} x_i \mid x \in T \} \mid \}$$

Les n+1 sommets de S^1 sont alors :

$(-\gamma, -\gamma, -\gamma, \ldots\ldots\ldots\ldots\ldots\ldots, -\gamma)$

$(n\gamma, -\gamma, -\gamma, \ldots\ldots\ldots\ldots\ldots\ldots, -\gamma)$

$(-\gamma, n\gamma, -\gamma, \ldots\ldots\ldots\ldots\ldots\ldots, -\gamma)$

$(-\gamma, -\gamma, n\gamma, \ldots\ldots\ldots\ldots\ldots\ldots, -\gamma)$

...

$(-\gamma, -\gamma, -\gamma, \ldots\ldots\ldots\ldots\ldots\ldots, -\gamma)$

<u>Calcul de $V(S^k)$, $k \geq 2$</u>

A chaque itération $k \geq 2$, le polytope S^k est défini à partir du polytope S^{k-1} auquel on ajoute une inégalité $l(x) \leq 0$ où l est une fonction affine définie par $l(x) = c^t x - c^t u^k$ ou bien $l(x) = (x - y^k)t^k + \delta h(y^k)$. L'ensemble des

sommets du polytope S^k, soit $V(S^k)$ est alors calculé par une des méthodes expliquées en détails au chapitre 2.

Calcul du point u^k

Le problème qui consisite à trouver un point u^k, intersection entre le segment $[z^k,x^k]$ et l'ensemble $\{ x \mid g(x)=0 \}$ se réduit à une équation d'une variable λ :

$$g(\lambda z^k + (1- \lambda)x^k) = 0 \qquad 0 \leq \lambda \leq 1$$

Il peut être résolu par exemple par la méthode de dichotomie ou par la méthode de Newton.

Calcul du sous-gradient

Pour calculer un élément t du sous-différentiel d'une fonction convexe h en un point x, on peut appliquer par exemple, la méthode qui consiste à choisir t sous la forme

$$t = \frac{h(x+ \varepsilon) - h(x- \varepsilon)}{2 \varepsilon}$$ où $\varepsilon > 0$ est un paramètre fixé.

1.4 Modifications de l'algorithme

L'algorithme implémentable tel qu'il est présenté au point 1.3 est certes efficace mais assez lent du point de vue de la convergence. Pour remédier à cela nous allons introduire quelques changements en vue de favoriser une convergence plus rapide vers la solution.

1.4.1 OPTIMISATION DU CALCUL DE z^K

Lorsqu'à l'itération k nous effectuons une coupe suivant la fonction objectif à partir d'un point u^k de T, il n'est pas nécessaire , à l'itération suivante de calculer z^{k+1} car $z^{k+1} = z^k$. En effet, plaçons-nous à l'itération k :

Nous disposons de z^k solution de $\begin{cases} \text{minimiser } c^t x \\ \text{s.c. } x \in V(S^k) \end{cases}$ ainsi que de

$u^k = [\, z^k, x^k \,] \cap \{ x \mid g(x)=0 \}$ où x^k est solution de $\begin{cases} \text{minimiser } g(x) \,. \\ \text{s.c. } x \in V(S^k) \end{cases}$

La coupe effectuée $l(x) = c^t x - c^t u^k$ définit $S^{k+1} = S^k \cap \{ x \mid l(x) \le 0 \}$.

Dès lors comme $c^t z^k < c^t u^k$, z^k sera toujours solution de $\begin{cases} \text{minimiser } c^t x \\ \text{s.c. } x \in V(S^{k+1}) \end{cases}$

1.4.2 RECHERCHE DE POINTS INTERIEURS A T

Nous avons vu que si à l'itération k, un point u^k tel que $h(u^k) \le 0$ et $g(u^k) = 0$ était disponible, nous pouvions effectuer une coupe suivant la

fonction objectif du type $l(x) = c^t x - c^t u^k \leq 0$. Ce sont les coupes les plus intéressantes à obtenir puisque,contrairement aux autres coupes du type $l(x) = (x-y^k)t^k + h(y^k)$, elles réduisent l'ensemble des points admissibles et ainsi favorisent la convergence de l'algorithme.

Dans le but d'obtenir à chaque itération de telles coupes, nous allons essayer de trouver des points qui appartiennent à T et qui annulent g : Pour ce faire, nous allons chercher un point pour lequel g est positive et un autre pour lequel g est négative . Dans la mesure où de tels points peuvent être trouvés, nous prenons l'intersection du segment qu'ils constituent avec la surface $g(x) = 0$. Les points d'intersection ainsi obtenus ne seront pris en compte que si ils appartiennent à T. Dans le cas où nous aurions retenus plusieurs points d'intersection, nous sélectionnerons celui pour lequel la fonction objectif est minimale en vue d'obtenir la meilleure coupe possible.

Méthode

A chaque itération, nous calculons z^k solution de $\begin{cases} \text{minimiser } c^t x \\ \text{s.c. } x \in V(S^k) \end{cases}$ et

x^k solution de $\begin{cases} \text{minimiser } g(x) \\ \text{s.c. } x \in V(S^k) \end{cases}$ (et ce dans *tous* les cas).

Nous allons essayer de trouver à partir de x^k un point $u_1{}^k$ correspondant aux exigences définies ci-dessus, et d'autre part,un point $u_2{}^k$ calculé à partir de z^k.

Recherche de $u_1{}^k$

Nous disposons de $x^k \in S^k$ pour lequel $g(x^k) \leq 0$. Il reste à trouver un

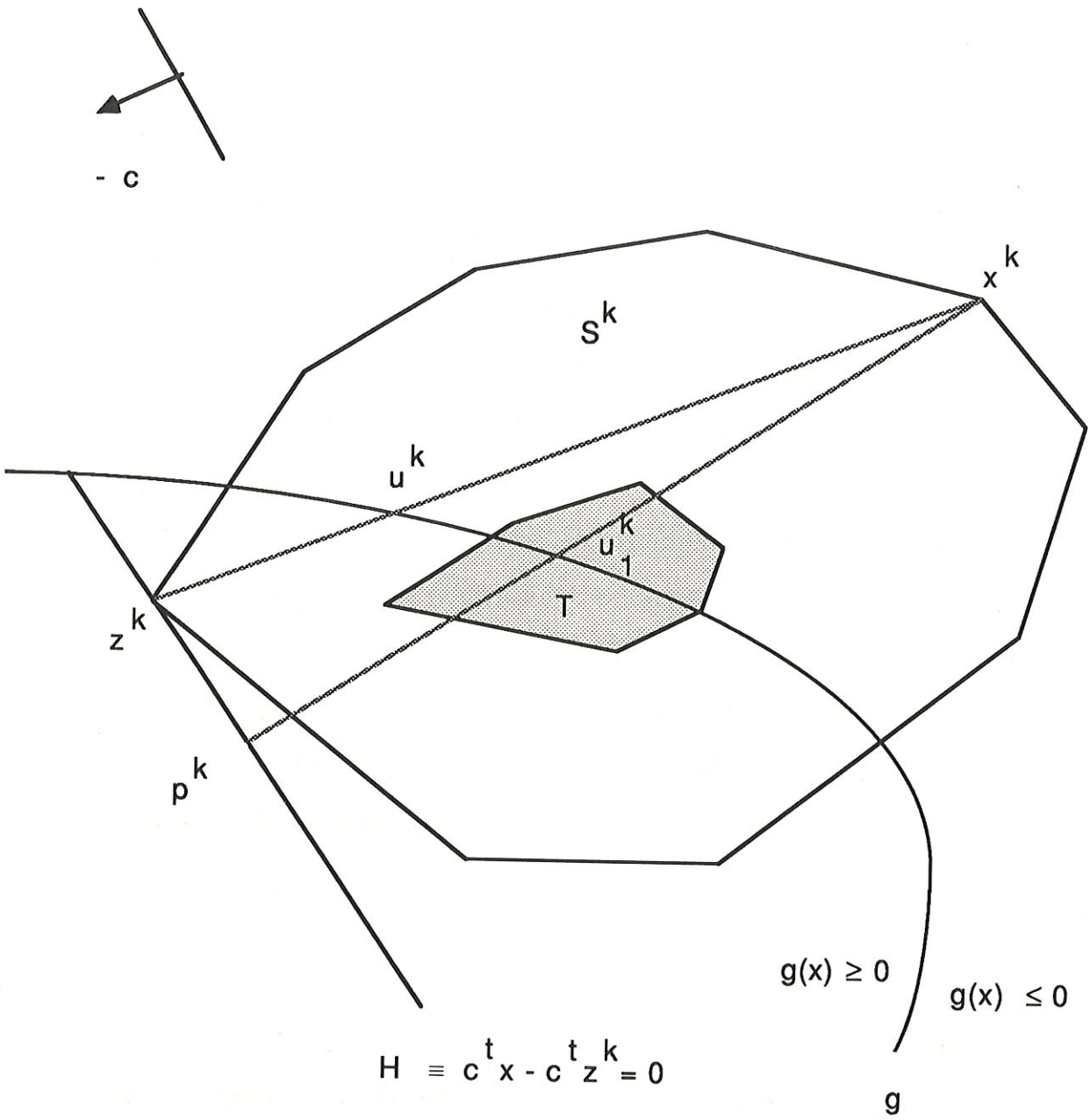


figure 13

point où g serait positive . Pour cela , en partant de x^k , déplaçons-nous dans la direction opposée du gradient de la fonction objectif jusqu'à rencontrer l'hyperplan passant par z^k et d'équation $c^tx-c^tz^k=0$ (on est sûr de le rencontrer car z^k minimise c^tx sur $V(S^k)$) . Notons p^k le point obtenu sur cet hyperplan (Il est à remarquer que p^k n'appartient pas forcément à S^k). Si $g(p^k) > 0$ alors le point $u_1{}^k$ est obtenu en effectuant l'intersection entre le segment $[x^k,p^k]$ et la surface $g(x) = 0$.
Si $h(u_1{}^k) \leq 0$, $u_1{}^k$ est admissible : nous le retenons ainsi que la valeur $h(u_1{}^k)$. [fig. 13]

Recherche de $u_2{}^k$

Cette fois nous partons du point z^k pour lequel $g(z^k)$ est soit >0, soit ≤0 . Nous nous déplaçons dans la direction du gradient de la fonction objectif jusqu'à rencontrer l'hyperplan passant par x^k et d'équation $c^tx-c^tx^k=0$.
Notons p^k le point obtenu sur cet hyperplan . Si $g(p^k)$ est du signe opposé à celui de $g(z^k)$, le point $u_2{}^k$ peut être calculé en prenant l'intersection entre le segment $[z^k,p^k]$ et la surface $g(x) = 0$. Si $h(u_2{}^k) \leq 0$, $u_2{}^k$ est admissible; nous le retenons ainsi que la valeur $h(u_2{}^k)$. [fig. 14]

Dans le cas où $u_1{}^k$ et $u_2{}^k$ ont été retenus, nous sélectionnons celui pour lequel la fonction objectif est minimale , notons le Pu . Si au cours de l'algorithme, nous trouvons un point $u^k \in T$, nous choisissons de nouveau entre u^k et Pu celui qui minimise la fonction objectif avant d'effectuer la

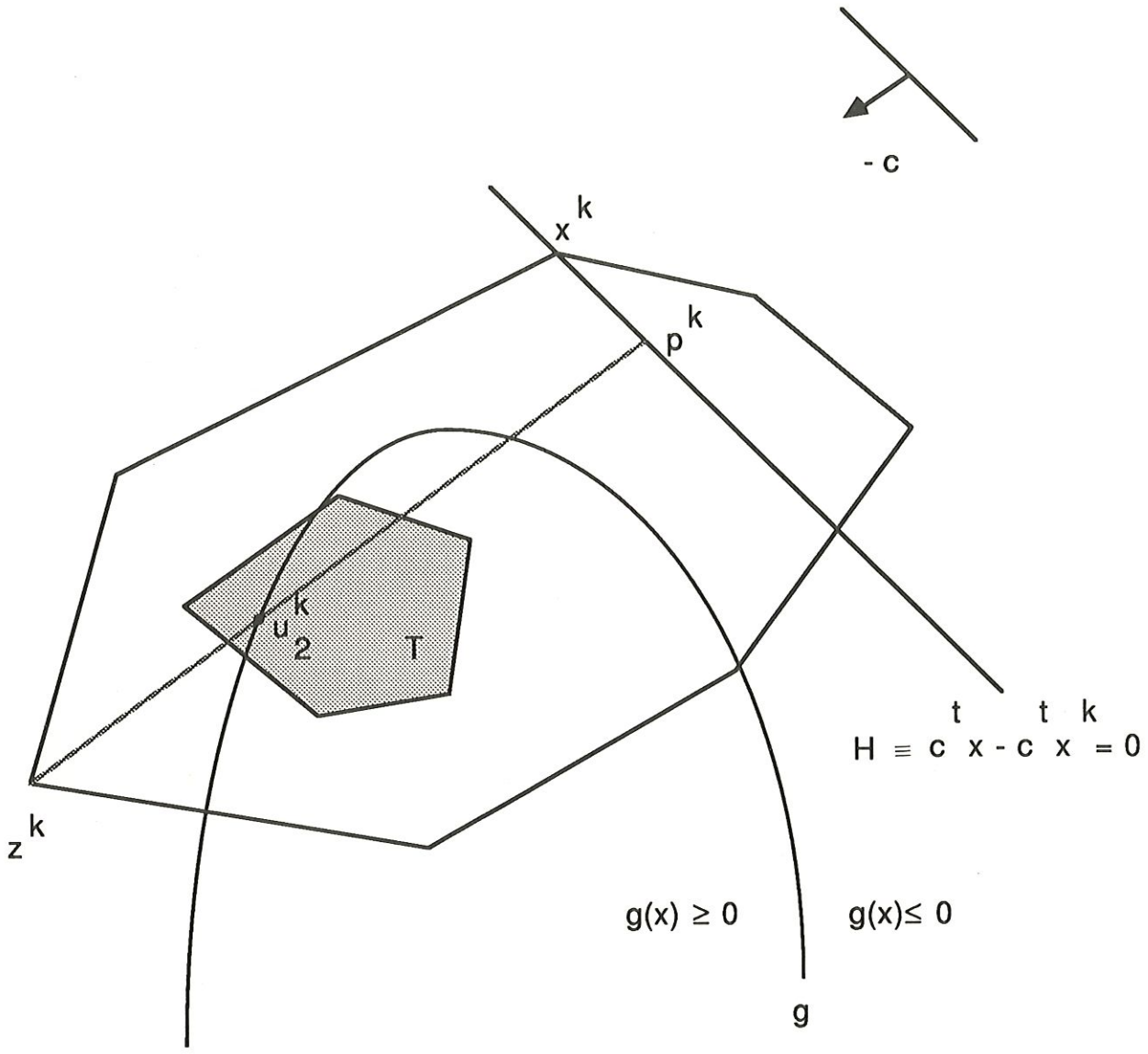


figure 14

coupe . Dans le cas où aucun point n'aurait été retenu, l'algorithme est poursuivi normalement.

1.4.3 AMELIORATION DE LA COUPE SUIVANT LA FONCTION OBJECTIF

Supposons qu'à l'itération k nous disposions d'un point u^k appartenant à T (c'est-à-dire $h(u^k) \leq 0$) tel que $g(u^k)=0$ et qui ne soit pas solution ($c^t z^k < c^t u^k$). Plutôt que d'effectuer une coupe en u^k, nous allons essayer à partir de ce point d'en trouver un autre appartenant à T pour lequel la fonction objectif aurait une valeur plus petite . En vue de diminuer le plus possible la valeur de la fonction objectif sans pour autant sortir de T, nous allons chercher un point sur la frontière de T. Dans ce but, puisque $h(u^k) \leq 0$ nous allons essayer de trouver un point où h est positive . Nous prendrons alors l'intersection du segment constitué par u^k et ce point avec la surface h(x)=0 . Le point d'intersection ainsi obtenu engendre une coupe qui réduit l'ensemble des points admissibles plus fortement que ne l'aurait fait la coupe à partir de u^k. De ce fait, la convergence de l'algorithme se trouve accélérée.

Méthode

A l'itération k nous disposons de z^k solution de $\begin{cases} \text{minimiser } c^t x \\ \text{s.c. } x \in V(S^k) \end{cases}$ et de

$u^k = [z^k, x^k] \cap \{ x \mid g(x)=0 \}$ avec $h(u^k) \leq 0$ où x^k solution de $\begin{cases} \text{minimiser } g(x) \\ \text{s.c. } x \in V(S^k) \end{cases}$

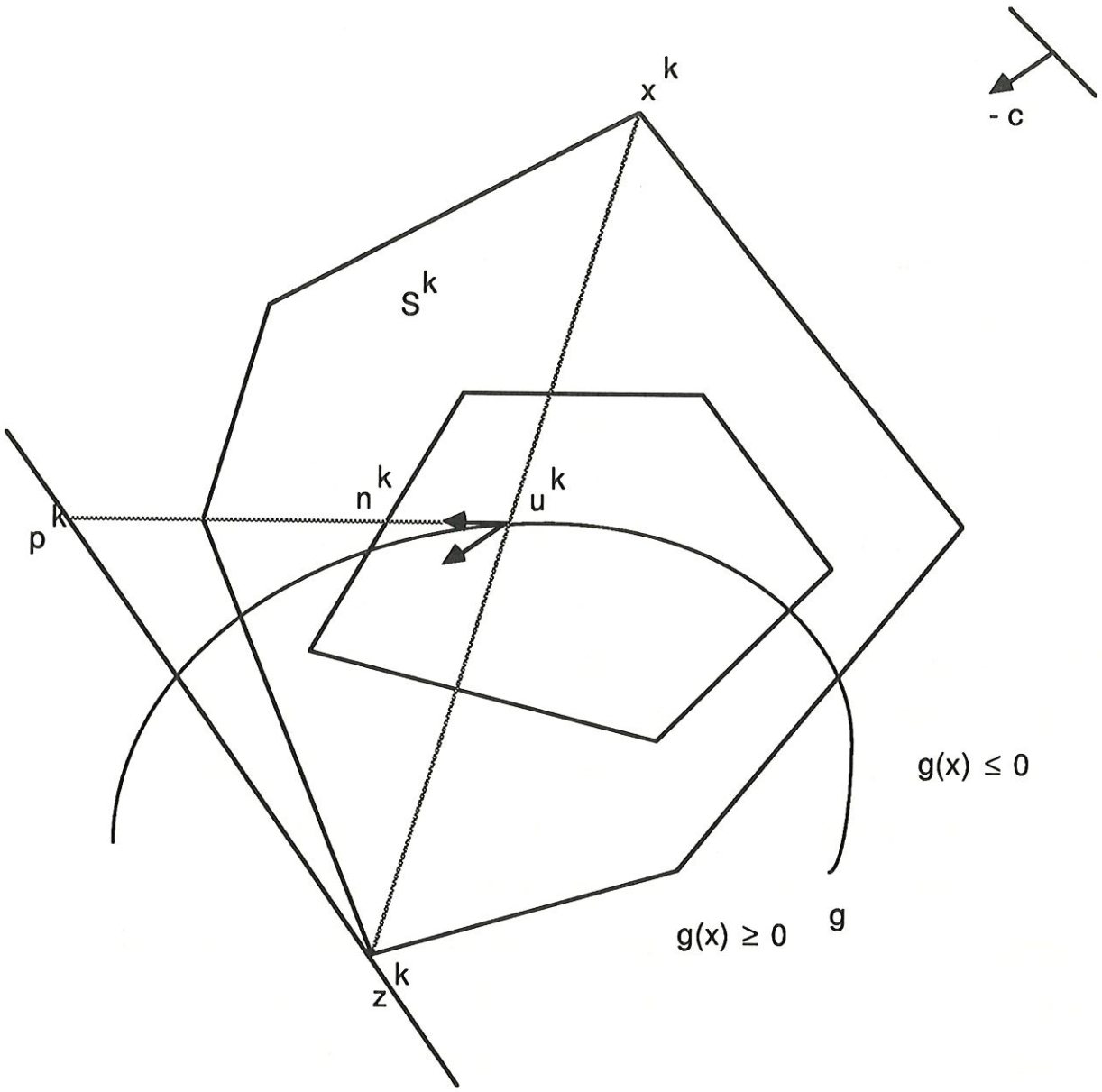


figure 15

Au point u^k, nous considèrons la direction opposée au gradient de la fonction objectif, projetée sur l'hyperplan tangent à g en u^k. En partant de u^k, nous nous déplaçons dans cette direction jusqu'à rencontrer l'hyperplan passant par z^k d'équation $c^t x - c^t z^k = 0$. Soit le point p^k obtenu sur cet hyperplan .

Si $h(p^k) > 0$, nous pouvons calculer n^k le nouveau point de T : $n^k = [u^k, p^k] \cap \{ x \mid h(x)=0 \}$. Nous effectuons alors une coupe passant par ce nouveau point , soit $l(x) = c^t x - c^t n^k$.

[fig. 15]

CHAPITRE 2 :

Recherche des nouveaux sommets et élimination des contraintes redondantes après une coupe.

2.1 Introduction.

Dans la méthode décrite au chapitre précédent, la partie la plus coûteuse en temps calcul est sans aucun doute la détermination de tous les nouveaux sommets du polytope S^{k+1} obtenus en coupant le polytope S^k par un hyperplan H.

Plusieurs méthodes ont été proposées pour résoudre ce problème, notamment par Falk et Hoffman [Réf.1] et par Thieu-Tam-Ban [Réf. 6] . Leur inconvénient est qu'elles ne sont performantes que pour des problèmes de petite taille.

Récemment Horst, de Vries et Thoai [Réf. 3] ont proposé une nouvelle méthode pour ce problème qui s'avére beaucoup plus efficace que les précédentes.

Dans ce chapitre, nous nous proposons d'exposer d'abord la méthode de Thieu-Tam-Ban [section 3.] et sa facon d'être implémentée dans le code de Thoai. Ensuite dans la section 4 , nous décrirons la méthode de Horst, de Vries et Thoai ainsi que la manière dont nous l'avons implémentée. Nous insisterons sur le cas où il y a "dégénérescence". Dans la section 5, nous indiquerons comment nous avons traité la redondance des contraintes. Les résultats numériques feront l'objet de la dernière section.

Avant de décrire les méthodes, commençons par formaliser le problème.

2.2 Position du problème .

Considérons un polytope compact S^k contenu dans R^n et défini par

$$S^k = \{ x \in R^n : g_i(x) = a^i x + b_i \leq 0 , i \in I \}$$

où I est un ensemble fini d'indices contenu dans N, $a^i \in R^n$ et

$b_i \in R$, $i \in I$.

Considérons un hyperplan H défini par $H = \{ x \in R^n : h(x) = \alpha x + \beta = 0 \}$

où $\alpha \in R^n$ et $\beta \in R$.

Soit P un des deux polytopes obtenus en coupant le polytope S^k par l'hyperplan H. Plus précisement on suppose que

$P = S^k \cap \{ x \in R^n : h(x) \leq 0 \}$.

Supposons que les sommets de S^k soient connus, et notons cet ensemble $V(S^K)$.

Notre problème consiste donc à déterminer l'ensemble des sommets de P, V(P).

Notons $C = S^k \cap H$

$$V^+(S^k) = \{ v \in V(S^k) : h(v) > 0 \}$$

$$V^-(S^k) = \{ v \in V(S^k) : h(v) < 0 \}$$

On voit immédiatement que C est un polytope et que $V(P) = V^-(S^k) \cup V(C)$.

Comme $V^-(S^k)$ peut être facilement obtenu, notre problème revient à déterminer l'ensemble V(C).

Proposition 1.

Avec les notations définies plus haut, on a : ω appartient à l'ensemble V(C) si et seulement si ω est ou bien un sommet de S^k se trouvant sur H,ou bien un point de la forme $[u,v] \cap H$, où $u \in V^-(S^k)$ et $v \in V^+(S^K)$.

2.3 Methode de Thieu-Tam-Ban.

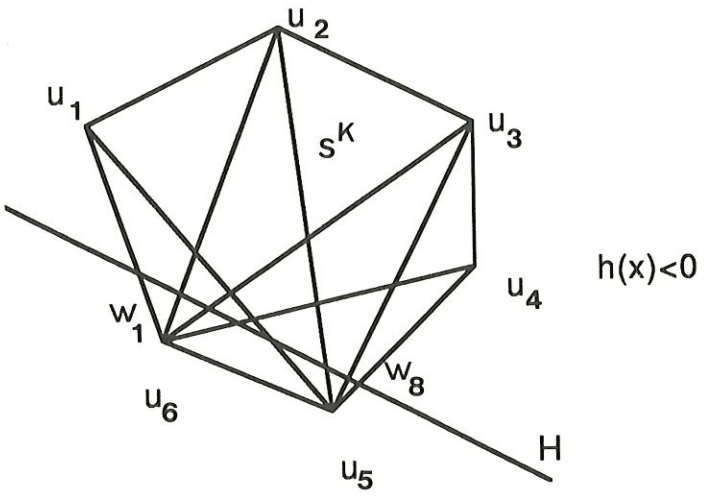
2.3.1. DESCRIPTION

Nous avons vu dans le paragraphe précedent, que le calcul de l'ensemble des nouveaux sommets, V(P), se ramenait à déterminer V(C), avec $C = S^K \cap H$

L'idée de la méthode de Thieu-Tam-Ban est de former tous les segments [u,v], où $u \in V^-(S^k)$ et $v \in V^+(S^k)$ et de calculer le point d'intersection ω de tous les segments [u,v] avec l'hyperplan H. Un test, basé sur le nombre de contraintes actives en ω , détermine alors le rejet ou le non rejet de ω comme nouveau sommet: si le nombre de contraintes actives en chacun des ω est plus grand ou égal à n-1, alors le point ω est un sommet.

2.3.2 EXEMPLE

Soit n = 2. Considérons le polytope S^k et la coupe H représentés ci-dessous.



On voit immédiatement que $V(S^k) = \{ u_1,u_2,u_3,u_4,u_5,u_6 \}$, que $V^-(P) = \{ u_1,u_2,u_3,u_4 \}$ et que $v^+(P) = \{ u_5,u_6 \}$.

Considérons tous les segments [u,v], où $u \in V^-(S^k)$ et $v \in V^+(S^K)$:

$[u_1, u_6]$, $[u_1, u_5]$, $[u_2, u_6]$, $[u_2, u_5]$, $[u_3, u_6]$, $[u_3, u_5]$,$[u_4, u_5]$, $[u_4, u_6]$.

Nous avons donc huit intersections $\omega_1 \omega_2 ... \omega_8$ .Pour chaque point trouvé, nous devons tester son admissibilité.

Examinons le point ω_1.

Il n'y a qu'une seule contrainte active en ω_1 :la contrainte passant par les sommets u_1 et u_6. ω_1 est donc un nouveau sommet.

En ce qui concerne le point ω_2, il n'existe pas de contraintes actives en ce point. ω_2 n'est donc pas un nouveau sommet.

Les autres points se traitent de la même façon.

Finalement on obtient $V(C) = \{ \omega_1, \omega_8 \}$ et $V(P) = \{ u_1, u_2, u_3, u_4, \omega_1, \omega_8 \}$

2.3.3 LIMITES DE LA METHODE

La méthode de Thieu-Tam-Ban s'avère efficace pour des problèmes de dimension peu élevée. En effet, le calcul de l'intersection de tous les segments [u,v], où $u \in V^-(S^k)$ et $v \in V^+(S^K)$, avec l'hyperplan H demande un effort remarquable. Notons, par exemple, que pour $\#(V^-(S^k)) = 200$ et $\#(V^+(S^K)) = 100$, nous devons calculer 20.000 segments [u,v].

2.4 Méthode de Horst, de Vries et Thoai .

2.4.1DESCRIPTION

Nous avons vu dans un paragraphe précédent, que le calcul de l'ensemble des nouveaux sommets V(P) se ramenait à déterminer V(C), où $C = P \cap H$.

L'idée générale de la méthode de Horst, de Vries et Thoai est basée sur les principes suivants :

Pour chaque sommet u appartenant à $V^-(S^k)$, notons par E(u) l'ensemble de toutes les droites passant par u et contenant un côté de S^k et notons par

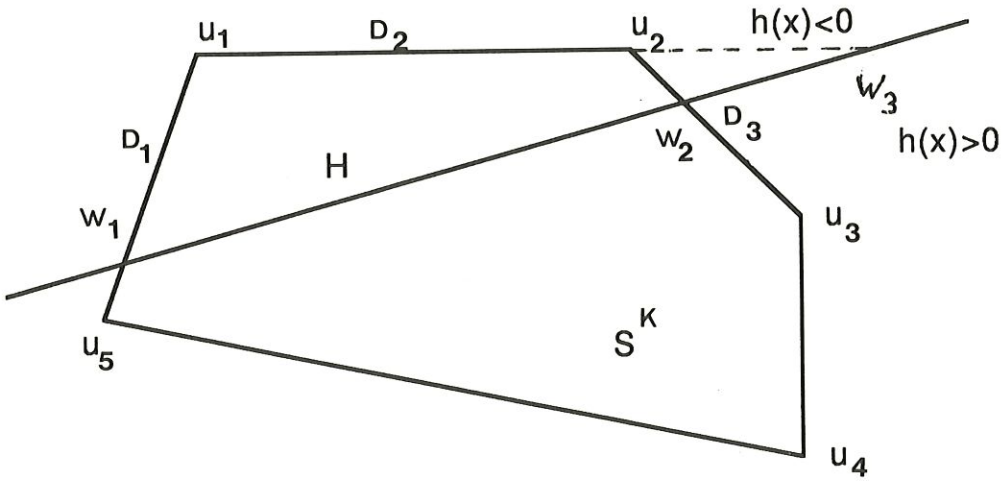
J(u) l'ensemble des indices des contraintes actives en u

ie $J(U) = \{ i \in I : g_i(u) = a^i u + b_i = 0 \}$

Par la proposition 1,nous avons que V(C) coincide avec l'ensemble des points obtenus en prenant les intersections , appartenant à S^k,des droites de E(u) avec l'hyperplan H ,et cela pour chaque sommet appartenant à $V^-(P)$.

2.4.2 Exemple.

Soit n = 2 .Considérons le polytope S^k et la coupe H représentés ci-dessous.



On voit immédiatement que $V(S^k) = \{ u_1, u_2, u_3, u_4, u_5 \}$,que $V^-(P) = \{ u_1, u_2 \}$ et que $v^+(P) = \{ u_3, u_4 \ u_5 \}$.

Examinons le sommet u_1. Nous avons que $E(u_1) = \{$ D1 ,D2 $\}$. Les intersections des droites D1 et D2 avec l'hyperplan H sont respectivement

ω_1 , ω_3 . Comme le point ω_3 n'appartient pas au polytope S^k,il ne peut être considéré comme un nouveau sommet.

De la même manière, nous pouvons dire que $E(u_2)$ = { D2, D3 },que les points d'intersection sont ω_2 et ω_3 et que le point ω_3 ne peut être considéré comme un nouveau sommet.

Finalement nous avons $V(C) = \{ \omega_2, \omega_1 \}$ et $V(P) = \{ u_1, u_2, \omega_2, \omega_1 \}$

REMARQUES

1. Comme on peut le constater, il arrive que l'on calcule plusieurs fois les mêmes intersections en partant de sommets différents dans un problème. Dans l'exemple considéré, on détermine deux fois le point ω_2. Dans l'implémentation de la méthode, il faudra donc éviter ce calcul inutile.

2. Dans la méthode présentée ci-dessus, nous avons calculé V(C) à partir de $V^-(S^k)$. Nous aurions très bien pu le faire à partir de $V^+(S^k)$.
En fait, nous choisissons $V^-(S^k)$ si $\# V^-(S^k) \leq \# V^+(S^k)$ et $V^+(S^k)$ autrement.

2.4.3 COMMENT DETERMINER LES INTERSECTIONS ?

Soit u un sommet de $V^-(S^k)$ (respectivement $V^+(S^k)$) .

Premier cas : supposons que u soit non dégénéré.

Dans ce cas, J(u) contient exactement les indices des n contraintes linéairement indépendantes g_i, actives en u. De plus, l'ensemble des inégalités $g_{ik}(x) = a^{ik} x + b_{ik} \leq 0$, $ik \in J(u)$ (k=1...n) définit un cône polyhédrique de sommet u.

Introduisons, pour chaque inégalité g_{ik} et pour la coupe $h(x)= \alpha x+\beta \leq 0$, une variable d'écart y_i.

Construisons le tableau du simplexe ci-dessous:

	X_1 X_2 ..... X_N	Y_1 Y_2 ... Y_N Y_{N+1}	Tind
Y_1	a_1^{i1} a_2^{i1} ..... a_N^{i1}	1 0 ..... 0 0	$-b_{i1}$
Y_2	a_1^{i2} a_2^{i2} ..... a_N^{i2}	0 0 ..... 0 0	$-b_{i2}$
Y_N	a_1^{iN} a_2^{iN} ..... a_N^{iN}	0 0 ..... 1 0	$-b_{iN}$
Y_{N+1}	α_1 α_2 ..... α_N	0 0 ..... 0 1	$-\beta$

où les n premières lignes correspondent aux égalités

$g_{ik}(x) = a^{ik} x + b_{ik} + y_i = 0$, $ik \in J(u)$ $(k=1...n)$ et la $(n+1)^{ème}$ à la coupe $h(x) = \alpha x + \beta + y_{n+1} = 0$. Cette dernière ligne représente les coûts relatifs aux variables x_1 x_2 .... x_n et y_1 y_2 .... y_{n+1} .En effet, nous pouvons considérer la coupe H comme fonction objectif.

Effectuons ensuite n pivotages de telle sorte que toutes les variables x_i deviennent des variables de base en prenant soin de ne pas faire sortir y_{n+1} de la base.

Nous constatons que le dernier tableau du simplexe est de la forme

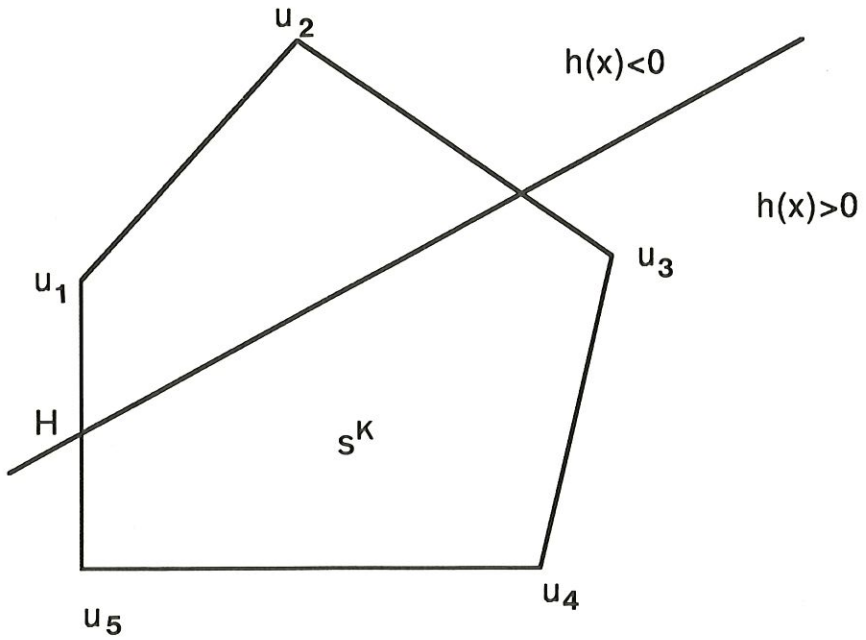
	x_1 x_2 x_N	Y_1 Y_2 Y_N	Y_{N+1}	RHS
X_1 X_2 X_N	T	D^{-1}		V
Y_{N+1}	S			

où D^{-1} représente l'inverse de la base, V le sommet u considéré et S les coûts relatifs.

Notre but est de déterminer les intersections de chaque côté de S^k passant par u avec la coupe. Il faut donc faire sortir la variable y_{n+1} de la base et faire entrer la variable y_i dans la base, y_i étant la variable associée au côté de S^k passant par u. Nous répétons cette opération autant de fois qu'il y a de cotés.

Le problème est de trouver les variables y_i qui doivent entrer dans la base, c'est-à-dire de déterminer les éléments de S surlesquels on va pivoter.

Supposons d'abord que nous travaillions avec $V^-(S^k)$ et soit u_j appartenant à $V^-(S^k)$.



Comme nous recherchons les intersections des côtes issus de u_j avec la coupe H, les côtés s'éloignant de H ne doivent pas être

considérés (par exemple, $u_1 u_2$ ne doit pas être pris en compte si nous partons de u_1).

Nous sommes donc amenés à examiner le problème

$$\begin{cases} \text{maximiser } h(\omega) \\ \text{s.c. } h(\omega) \leq 0 \\ \quad g_i(\omega) \leq 0,\ i \in J\ (u_j) \end{cases}$$

où $h(\omega)$ correspond à la coupe H et $g_i(\omega)$ aux côtés passant par u_j, et donc à regarder la dernière ligne du tableau comme une fonction objectif.

Par conséquent, comme nous savons que dans le tableau du simplexe, les colonnes de la matrice représentent les directions des côtés isssus de u_j, nous n'effectuerons un pivotage que sur les éléments strictement positifs de S.

Dans le cas où on travaille avec $V^+(S^k)$, on peut montrer par un raisonnement analogue que le pivotage doit se faire sur les éléments strictement négatifs de S.

Deuxième cas : supposons que u soit dégénéré.

Dans ce cas le nombre m de contraintes actives en u est tel que m est strictement plus grand que n.

Considérons tous les systèmes de n contraintes g_i actives en u, linéairement indépendantes. Ils sont au nombre de C_n^M. Nous pourrions alors regarder chacun des systèmes et appliquer l'algorithme présenté dans le premier cas. Nous trouverions ainsi l'ensemble de tous les nouveaux sommets. Mais ce procédé présente un inconvénient majeur: un sommet peut-être déterminé plusieurs fois. Nous nous trouvons donc devant une éventuelle répétition de certains sommets.

Pour éviter ce genre de problèmes, envisageons le moyen suivant. Considérons la coupe H(x) et n-1 contraintes actives en u. Le tout forme un système de n contraintes et ces systèmes sont au nombre de C_{N-1}^M. Dés lors notre problème consiste à trouver la solution de chacun des systèmes et à vérifier son admissibilité. Dans le cas où le point trouvé est admissible, un test permettra de détecter s'il appartient déjà à l'ensemble des nouveaux sommets, auquel cas nous le rejetons.

Nous devons donc résoudre un système de n équations à n inconnues. Pour cela, nous utiliserons la méthode de Gauss.

Plusieurs cas peuvent se présenter:

1. Le système considéré n'admet pas de solution. Géométriquement, ceci signifie qu'il n'y a pas d'intersection entre la coupe H et les n-1 côtés de S^k envisagés.

2. Le système considéré admet une solution unique. Géométriquement, l' hyperplan H et les n-1 côtés de S^k envisagés se coupent en un seul point.

2.4.4 COMMENT FORMER LE SYSTEME Q^K A PARTIR DU SYSTEME Q^{K-1}?

Soit u un sommet de $V^-(S^k)$ (respectivement de $V^+(S^k)$). Soit m le nombre de contraintes actives en u, avec $m > n$.

Chaque système Q^K est formé de n-1 contraintes actives en u et de la coupe H. Ces systèmes sont au nombre de C_{n-1}^{m} .

En outre, nous pouvons dire que le système Q^K diffère du système Q^{K-1} par une seule inégalité. Notre problème revient donc à remplacer la contrainte g_i par g_j ($i \neq j$) dans le système Q^{K-1}, en prenant garde à la redondance éventuelle avec des systèmes déjà construits. Pour cela, nous recherchons toutes les permutations de n-1 contraintes parmi m de telle sorte que la permutation j diffère de la permutation j-1 par une seule contrainte.

EXEMPLE

Soit u un sommet de $V^-(S^k)$. Supposons que n=3 et supposons qu'il y ait quatre contraintes actives en u: g_1, g_2, g_3, g_4. Nous avons immédiatement que les six permutations sont g_1g_2, g_1g_3, g_1g_4, g_2g_3, g_2g_4, g_4g_3. Nous devons alors les ordonner de sorte que la permutation j diffère de la permutation j-1 par une seule contrainte. Par exemple: g_1g_2, g_1g_3, g_1g_4, g_2g_4, g_2g_3, g_4g_3. Nous résoudrons donc d'abord le système $g_1g_2\,h(x)$, puis $g_1g_3\,h(x)$,..., $g_4g_3h(x)$.

Remarque.

Dans ce paragraphe, nous avons présenté une méthode pour déterminer les intersections de chaque côté du polytope S^k passant par le sommet considéré u avec la coupe. Notons ω ce nouveau point.

Dès lors, nous devons vérifier si ω est admissible, c'est-à-dire $g_i(\omega) = a^i \omega + b_i \leq 0$, $i \in I$. D'un point de vue pratique, il faut définir un seuil à partir duquel on accepte ω comme nouveau sommet, ce que nous avons fait. Cependant si on veut déterminer cette précision avec rigueur, une étude plus approfondie est nécessaire,mais ce n'est pas ici notre but.

2.4.5 ALGORITHME

Nous pouvons diviser l'algorithme en trois parties. La première porte sur le choix de $V^-(S^k)$ ou de $V^+(S^k)$. La deuxième détermine les nouveaux sommets et la troisième met à jour l' ensemble des sommets du nouveau polytope P.

Première partie : choix de $V^-(S^k)$ ou de $V^+(S^k)$.

ETAPE 1 On calcule le nombre de sommets u_i tel que premièrement $h(u_i)>0$, ensuite $h(u_i)<0$ et finalement $h(u_i)=0$.

ETAPE 2 Si les sommets u_i de S^k sont tels que ou bien $h(u_i)>0$ ou bien $h(u_i)=0$, alors la coupe H est un côté de S^k. Le polytope reste donc inchangé. Stop.

ETAPE 3 Si le nombre de sommets $u_i$tel que $h(u_i)<0$ est inférieur ou égal au nombre de sommets u_j tel que $h(u_j)>0$, alors on choisit $V^-(S^k)$,sinon on prend $V^+(S^k)$.

Deuxième partie : recherche des nouveaux sommet.

Soit u un sommet de $V^-(S^k)$ (respectivement de $V^+(S^k)$.

ETAPE 1 On calcule le nombre de contraintes actives en u.

ETAPE 2 On détermine si le sommet u est dégénéré ou non dégénéré. Pour cela,on teste si le nombre de contraintes actives en u est plus grand ou égal à n. Si le sommet est dégénéré, nous allons à l'étape 5 ,sinon à l'étape 3.

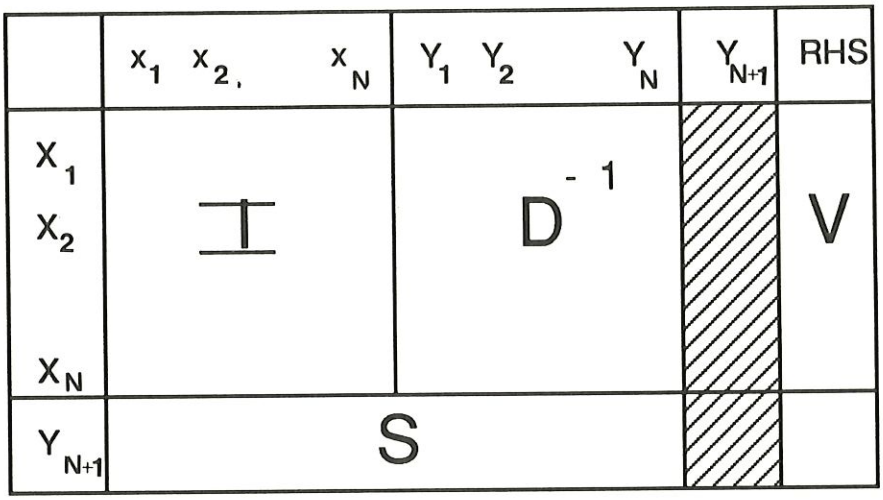
A. Cas où le sommet u est non dégénéré.

ETAPE 3 On calcule le tableau du simplexe aprés n pivotages

Au départ

	X_1 X_2 ..... X_N	Y_1 Y_2 Y_N Y_{N+1}	Tind
Y_1	a_1^{i1} a_2^{i1} ..... a_N^{i1}	1 0 ..... 0 0	$-b_{i1}$
Y_2	a_1^{i2} a_2^{i2} ..... a_N^{i2}	0 0 ..... 0 0	$-b_{i2}$
Y_N	a_1^{iN} a_2^{iN} ..... a_N^{iN}	0 0 ..... 1 0	$-b_{iN}$
Y_{N+1}	α_1 α_2 ..... α_N	0 0 ..... 0 1	$-\beta$

Après n pivotages.



Remarque.

Pour transformer le tableau, on ne fait pas explicitement les n pivotages. On utilise un autre procédé: on construit la matrice $\mathcal{D}$ (voir ci-dessus). On recherche son inverse . On calcule la variable d'écart,z,par z=-h(u). Enfin le vecteur S sera calculé: $(s_1\ s_2.\ldots\ldots s_n) = -(\alpha_1\alpha_2\ldots\ldots\alpha_n)\,D^{-1}$. Le vecteur V représente les composantes du sommet u.

ETAPE 4 On calcule les éventuels nouveaux sommets.

Pour chaque $S_j>0$,(respectivement $S_j<0$), on calcule le point d'intersection ω par $\omega_\kappa = \omega_\kappa - D_{kj}\,\frac{Z}{S_j}$

On teste l'admissibilité de ω .

B.Cas où le sommet est dégénéré.

ETAPE 5 Formation du système Q^k à partir du système Q^{k-1}.(voir 2.4.3)

ETAPE 6 Si le système Q^{k-1} n'admet pas de solution unique alors on détermine la solution de Q^k par le même procédé qu'en A. On teste ensuite son admissibilité et on vérifie que le point trouvé n'a pas déjà été déterminé. Il faut cependant faire attention au fait que ici nous n'avons pas nécessairement une matrice inversible. Si D n'est pas inversible, le système Q^k n'admet pas de solution et nous retournons à l'étape 5. Dans le cas inverse, on détermine la solution de Q^k en pivotant sur tout le tableau du "simplexe" correspondant au système Q^{k-1}.(Pour la détermination du pivot ,voir rem. 1). Si toutefois le pivot est nul, alors le système n'admet pas de solution et nous retournons à l'étape 5.

Troisième partie : mise à jour de V(P).

ETAPE 1 pour chaque sommet u_i,on vérifie que $h(u_i) \leq 0$. Si ce n'est pas le cas,on rejette le sommet.

Remarque

Détermination du pivot .

Nous sommes dans le cas où le système Q^{k-1} admet une solution unique et nous voulons déterminer la solution du système Q^k par un pivotage.

Supposons connues la permutation correspondant à Q^{k-1}, $(g_k)_{k \in \{1 \leq \ldots \leq i-1 \leq i < i+1 \leq \ldots \leq n-1\}}$ et celle correspondant à Q^k, $(g_k)_{k \in \{1 \leq \ldots \leq i-1 \leq j < i+1 \leq \ldots \leq n-1\}}$. Cette dernière permutation dera obtenue en faisant sortir la variable associée à la $i^{ème}$ contrainte et en faisant entrer la variable associée à la $j^{ème}$ variable.

2.5 Contraintes redondantes.

Dans ce paragraphe, nous proposons un critère pour éliminer les contraintes redondantes.

Considérons S^K un polytope défini par les inégalités $g_i(x) \leq 0$, $i \in I$, où I est un ensemble fini d'indices contenu dans N. La contraintes $g_k(x) \leq 0$, $k \in I$, est appelée redondante si et seulement si le polytope S^K reste inchangé losqu'on supprime cette contrainte.

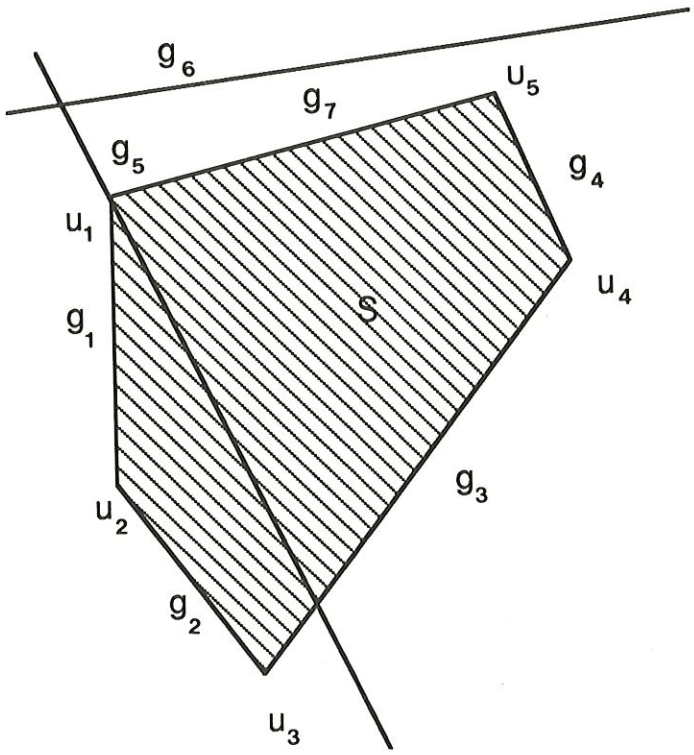
Critère d'élimination.

Considérons un polytope S^K contenu dans R^n et défini par les contraintes $g_i(x) \leq 0$, $i \in I$.

La contrainte $g_k(x) \leq 0$, $k \in I$, est redondante si et seulement si elle est active en un nombre de sommets strictement inférieur à n.

Exemple.

Soit n=2. Considérons le polytope défini par les contraintes g_1, g_2, g_3, g_4



On voit immédiatement que les contraintes g_5 et g_6 sont redondantes.

En effet, g_5 est active en un sommet u1 et g_6 ne passe par aucun sommet.

2.6 Exemples et résultats numériques.

2.6.1 CAS OU LE SOMMET EST NON DEGENERE

Considérons le problème suivant:

soit S^k un polytope défini par les contraintes

$$D1 \equiv x_2 - 4 \leq 0$$

$$D2 \equiv x_1 + x_2 \leq 0$$

$$D3 \equiv x_2 - 1 \leq 0$$

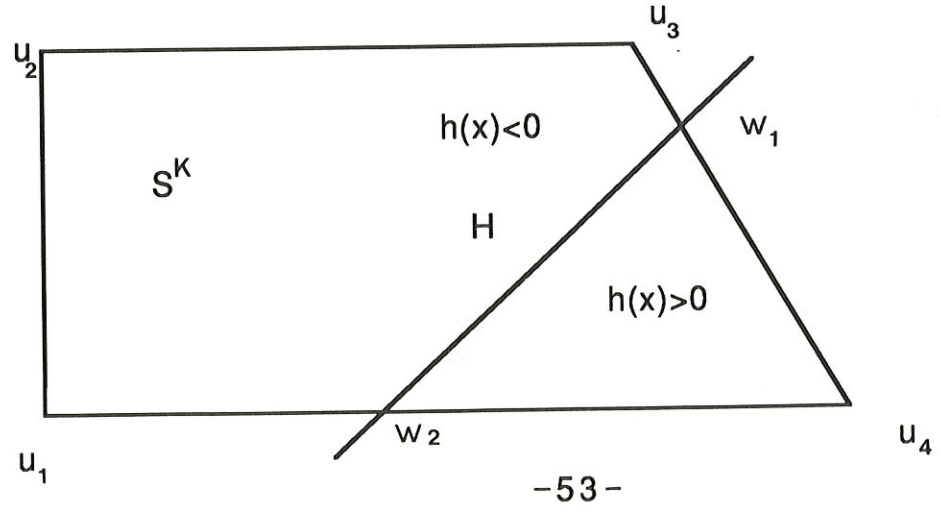
$$D4 \equiv -x_1 - 1 \leq 0$$

et soit $h(x) = -x_1 + x_2 + 3 \leq 0$ une coupe.

On peut voir facilement que l'ensemble des sommets du polytope S^k est $V(S^k) = \{ (1,1); (1,4); (4,4); (7,1) \}$. Le problème consiste à déterminer l'ensemble des sommets du nouveau polytope $P = S^k \cap \{ x \in R^2 : h(x) \leq 0 \}$.

Soient $y_1 \ldots\ldots y_5$ les variables d'écart correspondant aux contraintes D1...D4 et $h(x) \leq 0$.

Graphiquement,on a



RESOLUTION.

1. Déterminons $V^+(S^K)$ ou $V^-(S^K)$.

$h(u_1) \geq 0$; $h(u_2) \geq 0$; $h(u_3) \geq 0$; $h(u_4) \leq 0$. Nous choisissons donc $V^-(S^K)$.

Soit u_4 (7,1), le sommet de $V^-(S^K)$.

2.Calculons le nombre de contraintes actives en u_3.

D2,D3 sont actives en u_4 et donc u_4 est un sommet non dégénéré.

3.Recherchons les points d'intersection.

Pour cela,formons d'abord $D \begin{pmatrix} 0 & -1 \\ 1 & 1 \end{pmatrix}$

Nous pouvons dire que D^{-1} est de la forme $\begin{pmatrix} 1 & 1 \\ -1 & 0 \end{pmatrix}$

Calculons Z et S_j (j=1,2) : Z=3 ; S_1= 2; S_2= 1

Pour le cas S_1 = 2, nous obtenons un point d'intersection ω_1 de coordonnées

x_1=5.5 et x_2=2.5. ω_1 est admissible, donc un nouveau sommet.

Pour le cas S_2= 1, nous obtenons ω_2 de coordonnées x_1=4 et x_2=1. ω_2 est admissible, donc un nouveau sommet.

Nous avons finalement V(C)={ ω_1, ω_2 } et V(P)={ u_4 , ω_1, ω_2 }.

2.6.2.CAS OU LE SOMMET EST DEGENERE

Considérons le problème suivant:

soit S^k un polytope défini par les contraintes

$$\Pi 1 \equiv -12x_2 - X_3 + 84 \leq 0$$

$$\Pi 2 \equiv x_1 + 2x_3 - 12 \leq 0$$

$$\Pi 3 \equiv -4x + 14x_2 + 13x_3 - 169 \leq 0$$

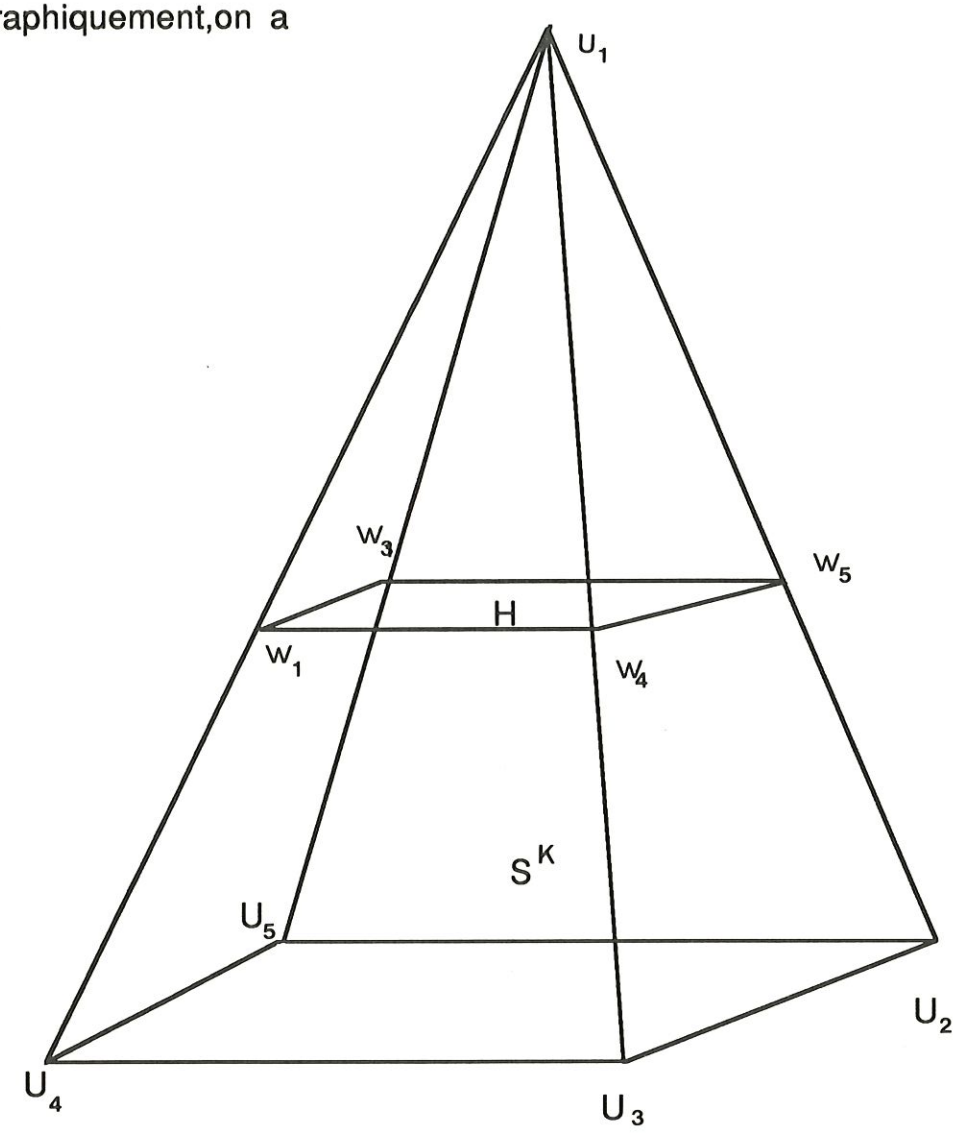
$$\Pi 4 \equiv -6x_1 - 5x_3 + 30 \leq 0$$

et soit $h(x) = - x_3 + 2 \leq 0$ une coupe.

On peut voir facilement que l'ensemble des sommets du polytope S^k est $V(S^k)$ ={ (0,6.5,6); (12,7,0); (12,15.5,0); (5,7,0); (5,13.5,0)}. Le problème consiste à déterminer l'ensemble des sommets du nouveau polytope $P = S^k \cap \{ x \in R^3 : h(x) \leq 0 \}$. Soient y_4.....y_8 les variables d'écart

correspondant aux contraintes $\Pi 1 \ldots \Pi 4$ et $h(x) \leq 0$.

Graphiquement,on a



RESOLUTION.

1. Déterminons $V^+(SK)$ ou $V(S^K)$.

$h(u_1) < 0$; $h(u_2) > 0$; $h(u_3) > 0$; $h(u_4) > 0$. $h(u_5) > 0$; Nous choisissons donc $V^-(S^K)$.

Soit u_1 (0,13.5,6), le sommet de $V^-(S^K)$.

2.Calculons le nombre de contraintes actives en u_3.

$\Pi 1, \Pi 2, \Pi 3, \Pi 4$ sont actifs en u_1 et donc u_1 est un sommet dégénéré.

3.Recherchons les points d'intersection.

A. Nous formons le 1er système: $\Pi 1, \Pi 2$,h(x).

Construisons la matrice D $\begin{pmatrix} 0 & -12 & -1 \\ 1 & 0 & 2 \\ 0 & 0 & -1 \end{pmatrix}$ et le vecteur Tind $\begin{pmatrix} -84 \\ 12 \\ -2 \end{pmatrix}$

La matrice D^{-1} sera $\begin{pmatrix} 0 & 1 & 2 \\ -1/12 & 0 & 1/12 \\ 0 & 0 & -1 \end{pmatrix}$

Le nouveau point ω_1 sera donc

$$\begin{pmatrix} 0 & 1 & 2 \\ -1/12 & 0 & 1/12 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} -84 \\ 12 \\ -2 \end{pmatrix} = \begin{pmatrix} 8 \\ 41/6 \\ 2 \end{pmatrix}$$

Comme ω_1 vérifie toutes les contraintes, il est un nouveau sommet. Calculons et formons le tableau du"simplexe".

	Y_4	Y_5	Y_6	Y_7	Y_8	b
X_1	0	1	0	0	2	8
x_2	-1/12	0	0	0	1/12	41/6
Y_5	7/6	4	1	0	119/6	238/6
Y_7	0	6	0	1	7	28
X_3	0	0	0	0	-1	2

B. Formons le deuxième système: $\Pi 1$, $\Pi 3$, h(x).

Le pivot correspond à l'élément (3,2) du tableau. Comme cet élément (=4) est non nul,nous effectuons un pivotage sur tout le tableau. Le tableau précédent devient alors:

	Y_4	Y_5	Y_6	Y_7	Y_8	b
X_1	-7/24	0	-1/4	0	-7/24	-71/6
X_2	-1/12	0	0	0	1/12	41/6
Y_5	7/24	1	1/4	0	119/24	238/12
Y_7	-7/4	0	-3/2	1	139/4	-91
X_3	0	0	0	0	-1	2

Nous avons ainsi ω_2 de coordonnées $X_1 = -71/6$; $X_2 = 41/6$; $X_3 = 2$.

Comme ce point ne vérifie pas les contraintes, il n'est pas un sommet.

C. Formons le troisième système: $\Pi 1$, $\Pi 4$,h(x).

Le pivot correspond à l'élément (4,3) du tableau. Comme cet élément (=-3/2) est non nul,nous effectuons un pivotage sur tout le tableau. Le tableau précédent devient alors:

	Y_4	Y_5	Y_6	Y_7	Y_8	b
X_1	0	0	0	1/6	7/6	20/6
X_2	-1/12	0	0	0	1/12	41/6
X_5	0	1	0	-1/6	-5/6	14/3
X_6	7/6	0	1	-2/3	139/6	182/3
X_3	0	0	0	0	-1	2

Nous avons ainsi ω_3 de coordonnées $X_1 = 20/6$; $X_2 = 41/6$; $X_3 = 2$.

Comme ce point vérifie les contraintes et est différent des points trouvés précédemment, il est un nouveau sommet.

D. Formons le quatrième système: $\Pi 2$, $\Pi 4$, h(x).

Le pivot correspond à l'élément(3,1) du tableau. Comme cet élément est nul, le système n'admet pas de solution.

E. Formons le cinquième système: $\Pi 2$, $\Pi 3$, h(x) .

Construisons la matrice D $\begin{pmatrix} 1 & 0 & 2 \\ -4 & 14 & 13 \\ 0 & 0 & -1 \end{pmatrix}$ et le vecteur Tind $\begin{pmatrix} 12 \\ 169 \\ -2 \end{pmatrix}$

La matrice D^{-1} sera $\begin{pmatrix} 1 & 0 & 2 \\ 2/7 & 1/14 & 3/2 \\ 0 & 0 & -1 \end{pmatrix}$

Le nouveau point ω_4 sera donc

$$\begin{pmatrix} 1 & 0 & 2 \\ 2/7 & 1/14 & 3/2 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} 12 \\ 169 \\ -2 \end{pmatrix} = \begin{pmatrix} 8 \\ 25/2 \\ 2 \end{pmatrix}$$

Comme ω_4 vérifie toutes les contraintes et est différent des points trouvés avant, il est un nouveau sommet.

Formons le tableau du"simplexe".

	Y_4	Y_5	Y_6	Y_7	Y_8	b
X_1	0	1	0	0	2	8
X_2	0	2/7	1/14	0	3/2	25/2
Y_4	1	24/7	6/7	0	17	68
Y_7	0	6	0	1	7	28
X_3	0	0	0	0	-1	2

E. Formons le sixième système: $\Pi 3$, $\Pi 4$,h(x).

Le pivot correspond à l'élément (4,2) du tableau. Comme cet élément (=6) est non nul,nous effectuons un pivotage sur tout le tableau. Le tableau précédent devient alors:

	Y_4	Y_5	Y_6	Y_7	Y_8	b
X_1						20/6
X_2						67/6
Y_4						
Y_5						
X_3						2

Comme ω_5 vérifie toutes les contraintes et est différent des points trouvés auparavant, il est un nouveau sommet.

4.MISE A JOUR DE L'ENSEMBLE DES SOMMETS.

Comme $h(u_2),h(u_3),h(u_4),h(u_5)$ sont positifs, nous devons les rejeter.

Nous avons donc $V(P) = \{ u_1,\omega_1,\omega_3,\omega_4,\omega_5 \}$

2.6.3 Résultats numériques.

Nous avons implémenté en FORTRAN 77 sur VAX 8600 la méthode de Horst, de Vries et de Thoai. Cette dernière ainsi que la méthode de Thieu-Tam-Ban ont été testées sur plusieurs problèmes .

Dans chacun des essais ,le polytope S^k est représenté par un polygone régulier à m côtés, la coupe est définie par la droite D d'équation $x - 4.5 \leq 0$.

Table des variables.

n : dimension de S^k.

m : nombre de contraintes définissant S^k.

$V(S^k)$: nombre de sommets du polytope S^k.

V(P) : nombre de sommets du polytope P, obtenu en coupant S^k par un hyperplan H.

T1 : temps cpu pour notre algorithme.

T2 : temps cpu pour la procédure de Thieu-Tam-Ban.

Comparaison de notre algorithme avec la procédure de Thieu-Tam-Ban.

m	n	V (S)	V (S)	T1 [sec.]	T2 [sec.]
100	2	15	85	0.07	1.53
250	2	35	215	0.24	22.79
500	2	71	429	0.86	190.25
750	2	107	643	1.85	666.23
1000	2	143	857	3.14	1574.14
1250	2	179	1071	4.82	3113.85
1500	2	215	1285	6.90	5444.39

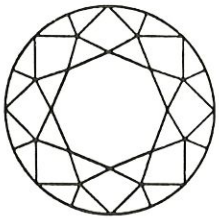
Par ces exemples, on constate aisément que la méthode que nous avons implémentée donne lieu à de meilleurs résultats. En effet, la différence de temps CPU entre les deux méthodes est considérable même pour un nombre élevé de sommets.

CHAPITRE 3 :

Application à un problème de découpe de diamants.

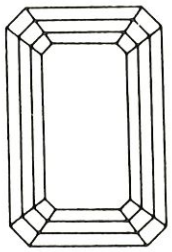
3.1 Introduction

Le problème de découpe d'une pierre brute pour en faire un ou plusieurs diamants est un problème très complexe. La façon de procéder dépend d'une part des propriétés géométriques et physiques de la pierre brute et d'autre part de la forme du produit fini désiré. La figure [1] ci-dessous donne les formes de pierres taillées les plus courantes :



LE BRILLANT
la plus classique : c'est une taille ronde, à 57 facettes.

LA MARQUISE
OU NAVETTE
taille oblongue : sa forme est gracieuse.



LA TAILLE ÉMERAUDE
rectangulaire et à degrés sur les côtés et les angles.

LA POIRE
ou " goutte d'eau " dont les feux sont aussi étincelants que ceux du brillant.

LA BAGUETTE
taille rectangulaire longue et étroite qui est une pierre d'accompagnement de forme particulièrement élégante.

figure 1.

Très souvent le produit fini désiré sera de forme "brillant". Cette forme a été conçue pour donner au diamant une diffusion optimale de la lumière. Les tolérances sur les différents angles et sur les dimensions relatives sont extrêmement étroites.

Comme la valeur de la pierre taillée dépend de son poids,de sa pureté et de sa couleur, il est très important pour la taille, de connaître avec précision la forme de la pierre brute.

Pour le moment, la taille des diamants est confiée à un expert qui ne dispose pas d'assistance scientifique. Les méthodes utilisées sont heuristiques et basées uniquement sur l'expérience et l'habileté de l'expert. Cependant vu les sommes énormes qui sont en jeu, un outil scientifique capable d'aider les experts dans leur tâche serait le bienvenu.

Une première approche de ce problème a été présentée dans [Réf.5] Elle permet de trouver dans une pierre brute convexe ou non, le plus gros diamant d'une forme spécifique donnée, l'orientation étant fixée a priori. Dans ce chapitre on désire présenter une solution au même prorblème mais où l'orientation n'est pas fixée a priori. On se limitera toutefois au cas simple d'une pierre brute convexe.

Après avoir modélisé le problème sous la forme d'un programme d.c. (section 3.2), nous présenterons dans la section 3.3 les résultats numériques ainsi que les graphiques obtenus.

3.2 Le problème à deux dimensions

Le but étant de maximiser la valeur totale de la pierre taillée, le problème peut s'exprimer de la façon suivante :
connaissant la forme d'une pierre brute, comment y tailler un ou plusieurs diamants d'une forme spécifiée à l'avance pour que la valeur de ceux-ci soit la plus grande possible. C'est typiquement un problème d'optimisation à condition de pouvoir exprimer de façon mathématique l'objectif à maximiser ainsi que les contraintes à respecter.

En vue de réduire la complexité du problème on considère dans une première étude les simplifications suivantes :

- un seul diamant de forme spécifiée doit être taillé dans la pierre brute convexe.
- la valeur d'un diamant est uniquement basée sur son poids ou, ce qui revient au même, son volume.
- la pierre brute ne comporte pas d'impureté.

De plus on suppose pour l'instant que ce problème est à deux dimensions.
En effet l'étude du cas bidimensionnel constitue une excellente préparation
à l'étude du problème général.

3.1.1 Position du problème

Dans le plan R^2 =xOy, on se donne un polygône convexe P_0 (= pierre brute). Soit r le nombre de ses "faces" (= côtés). Ce nombre r est égal au nombre de "points extrêmaux" (= sommets). Soit d'autre part un polygône

convexe P_R (= pierre de référence) de forme donnée dans le plan. Soit s le nombre de ses sommets.

3.1.2 Question

Comment inscrire dans le polygône convexe fixé P_0 un polygône convexe P_T le plus grand possible et de même forme que P_R ?

3.1.3 Solution proposée

1) Définitions et notations

Désignons par (x_j^R, y_j^R) les coordonnées cartésiennes des sommets de P_R, j=1,...s numérotés dans le sens des aiguilles d'une montre. Appelons P un polygône convexe de sommets (x_j, y_j) où j=1,2,...s qui sont numérotés dans le même sens que P_R .

On dira que P a la même forme que P_R si et seulement si on peut trouver $d \geq 0$, $\theta \in]0,2\pi[$, $p \in R$, $d \in R$ tels que

$$\forall\ j=1,2,...s \quad (x_j,y_j,1) = (x_j^R, y_j^R, 1) \begin{pmatrix} d\cos\theta & d\sin\theta & 0 \\ -d\sin\theta & d\cos\theta & 0 \\ p & q & 1 \end{pmatrix} \quad (1)$$

où d est le facteur de dilatation, θ l'angle de rotation, p la translation le long de l'axe des x et q la translation suivant l'axe des y. Ces facteurs sont les mêmes pour chaque sommet.

2) Description de la fonction objectif

Soit P_T (qui sera le diamant taillé) le polygône de sommets (x_j,y_j) j=1,2,...s de surface S_T, de même forme que P_R et inscrit dans P_0. Notons S_0 la surface de P_0. Le problème d'optimisation qui nous intéresse est alors le suivant :

Trouver (x_j , y_j) ***j=1,...s tels que la surface résiduelle*** $(S_0 - S_T)$ ***soit minimale.***

Or comme S_0 est donné, il suffit, pour minimiser $(S_0 - S_T)$, de maximiser S_T.

La surface S_T est très facile à calculer ; elle est donnée par la formule suivante :

$$S_T = \frac{1}{2} \sum_{j=1}^{s} (x_j y_{j+1} - x_{j+1} y_j) \qquad (2)$$

où l'on a posé $(x_{s+1} , y_{s+1}) = (x_1 , y_1)$. Les sommets ont été numérotés de façon à ce que S_T soit positive.

3)contraintes

En ce qui concerne les contraintes, elles peuvent s'exprimer de la

manière suivante :

1ère contrainte : P_T est inscrit dans P_0 :

Désignons par $a_i x + b_i y + c_i = 0$ $(i=1,...r)$, les équations des r droites qui déterminent la frontière de P_0 ; Supposons que P_0 est déterminé par les r inéquations :

$$a_i x + b_i y + c_i \leq 0 \qquad i=1,...r$$

Les contraintes s'écrivent alors

$$a_i x_j + b_i y_j + c_i \leq 0 \qquad i=1,..r \text{ et } j=1,...s \qquad (3)$$

puisque P_0 est convexe.

2ème contrainte : P_T a la même forme que P_R

Elle s'exprime par :

$$\forall\, j=1,2,...s \quad (x_j, y_j, 1) = (x_j^R, y_j^R, 1) \begin{pmatrix} d\cos\theta & d\sin\theta & 0 \\ -d\sin\theta & d\cos\theta & 0 \\ p & q & 1 \end{pmatrix} \qquad (4)$$

En portant les valeurs de x_j et y_j données par la relation (4) dans l'expression (2), nous obtenons l'expression suivante

$$\underset{d,\theta,p,q}{\text{Maximiser}} \ \frac{1}{2} \sum_{j=1}^{s} (x_j^R y_{j+1}^R - x_{j+1}^R y_j^R)\, d^2$$

Or comme $\frac{1}{2} \sum_{j=1}^{s} (x_j^R y_{j+1}^R - x_{j+1}^R y_j^R)\, d^2 = S_R$

le problème revient à $\begin{cases} \text{Maximiser} \\ d,\theta,p,q \end{cases} d^2$

Portons maintenant les valeurs de x_j et y_j dans l'expression (2) :

$C_{ij}(d,\theta,p,q) \equiv$

$$(a_i x_j^R + b_i y_j^R)\, d \cos\theta + ((b_i x_j^R + a_i y_j^R)\, d \sin\theta + a_i p + b_i q + c_i \leq 0$$

Le problème d'optimisation est alors le suivant:

$$\begin{cases} \text{Maximiser } d^2 \\ d,\theta,p,q \\ \\ \text{s. c.} \quad C_{ij}(\, d,\theta,p,q\,) \leq 0 \quad i=1,....r \text{ et } j=1,...s \\ \\ 0 \leq \theta \leq 2\pi \end{cases}$$

C'est un problème d'optimisation **non linéaire** à 4 variables et à rs contraintes **non linéaires**, θ étant une variable bornée.

Il serait intéressant d'avoir toutes les contraintes linéaires, car le problème serait plus facile à résoudre. Pour cela, posons $u = d \cos\theta$ et $v = d \sin\theta$.

L'équation (4) devient alors :

$$\forall\ j=1,2,...s \quad (x_j, y_j, 1) = (x_j^R, y_j^R, 1) \begin{pmatrix} u & v & 0 \\ -v & u & 0 \\ p & q & 1 \end{pmatrix} \qquad (5)\ ,$$

et on a le problème de maximisation suivant à résoudre :

$$\begin{cases} \text{Maximiser } u^2 + v^2 \\ u,v,p,q \\ \text{s.c.} \quad Q_{ij}(u,v,p,q) \leq 0 \qquad i=1,\ldots r \text{ et } j=1,\ldots s \end{cases}$$

où $Q_{ij}(u,v,p,q) \equiv$

$$(a_i x_j^R + b_i y_j^R)\, u + (\, (b_i x_j^R + a_i y_j^R)\, v + a_i p + b_i q + c_i \leq 0$$

C'est un problème **non linéaire** à 4 variables et rs contraintes **linéaires**.

Il reste à mettre ce problème sous forme d.c. convexe canonique pour pouvoir le traiter par l'algorithme exposé dans les chapitres précédents. Nous sommes alors amené à résoudre le problème suivant :

$$\begin{cases} \text{Minimiser } -t \\ u,v,p,q,t \\ \text{s.c.} \quad Q_{ij}(u,v,p,q) \leq 0 \qquad i=1,\ldots r \text{ et } j=1,\ldots s \\ \qquad\quad t - (u^2 + v^2) \leq 0 \end{cases}$$

C'est un problème d.c. car la dernière contrainte est anti-convexe.

3.3 Résultats numériques

Nous avons implémenté en FORTRAN 77 sur l'ordinateur VAX 8600, la méthode de recherche du diamant optimal pouvant être taillé dans une pierre brute donnée (cfr. annexe).

Un nombre assez important de tests ont été effectués avec succès. On indiquera dans les pages qui suivent les résultats de quelques uns d'entre eux avec les données correspondantes.

Pour donner une idée de l'efficacité des modifications que nous avons apportées à l'algorithme de Thoai, nous allons comparer la version originale avec la version modifiée où nous calculons les sommets de façon différente en tenant compte de la redondance et où nous cherchons d'autres points pour favoriser de meilleures coupes.

Il est à noter que dans les tableaux qui suivront, S_{max} représente le polytope engendré par l'algorithme qui a le nombre maximum de sommets tandis que S_{final} représente le polytope engendré à la dernière itération.

La dimension de l'espace dans lequel nous travaillons est toujours de 5 puique le problème que l'on résout est donné par

$$\begin{cases} \underset{u,v,p,q,t}{\text{Minimiser}} \; -t \\ \text{s.c.} \quad Q_i(u,v,p,q) \leq 0 \qquad i=1,\dots r \quad i=1,\dots s \\ \qquad\quad t^2 - (u^2+v^2) \leq 0 \qquad\qquad \text{(voir § 3.2)} \end{cases}$$

La variable d'écart t introduite pour linéariser la fonction objectif entraine une convergence assez lente des algorithmes étant donné la nature des coupes et le fait que cette variable t n'intervienne que dans la

contrainte anticonvexe. Dès lors les variables u,v,p et q sont parfois correctes bien avant que t ne soit ajusté. C'est ce qui explique d'une part le nombre assez élévé d'itérations (quelques vingt itérations) pour les tests assez simples et d'autre part que la méthode de Thoai n'aboutit à aucune solution "complète", le nombre de sommets dépassant parfois la limite maximale prévue.

TEST 1

A. Le diamant brut P_0 (en rouge) est défini par les inégalités suivantes :

1. $-0.25x + y - 5 \leq 0$
2. $0.25x + y - 5 \leq 0$
3. $2x - y - 4 \leq 0$
4. $-2x - y - 4 \leq 0$

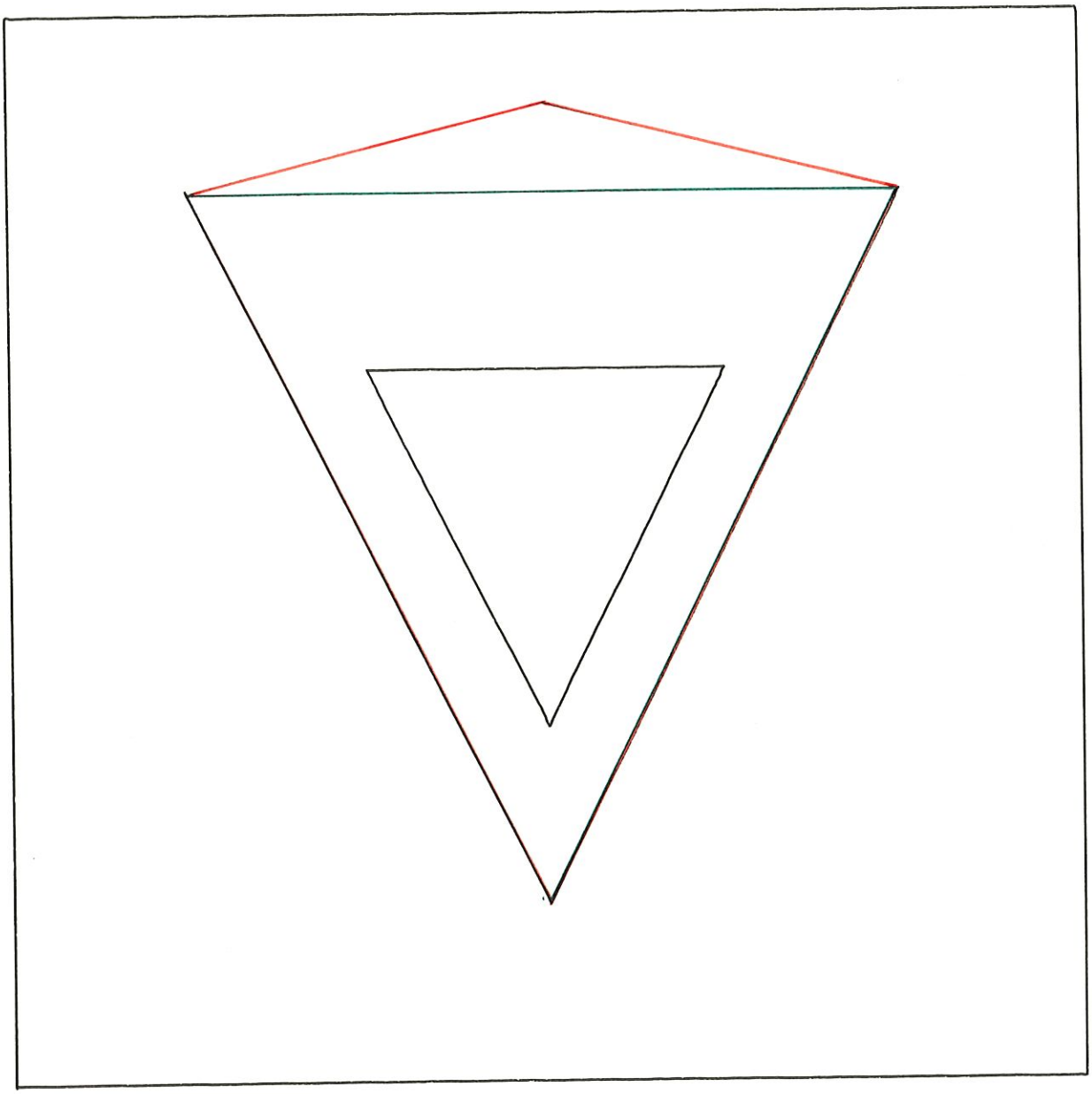
B. Le diamant de référence P_R (en noir) est déterminé par ses sommets :

1. (0,-2)
2. (-2, 2)
3. (2, 2)

C. Tableau des résultats :

	Algorithme de Thoai	Algorithme modifié
Nombre d'itérations	16	15
Nombre de contraintes	21	11
$V(S_{final})$	290	37
$V(S_{max})$	290	37
Temps CPU (sec.)	2.81	0.38

L'efficacité de la nouvelle façon de calculer les sommets en éliminant les contraintes redondantes apparaît ici nettement. Nous constatons par ailleurs que le fait d'engendrer de nouvelles coupes fait gagner une itération.



DILATATION = 2

TEST 2

A. Le diamant brut P_0 (en rouge) est défini par les inégalités suivantes :

1. $x - y$
2. $2x + y - 4 \leq 0$
3. $3x - y - 6 \leq 0$
4. $-y - 6 \leq 0$
5. $-2x - y - 8 \leq 0$
6. $-2x + y - 4 \leq 0$

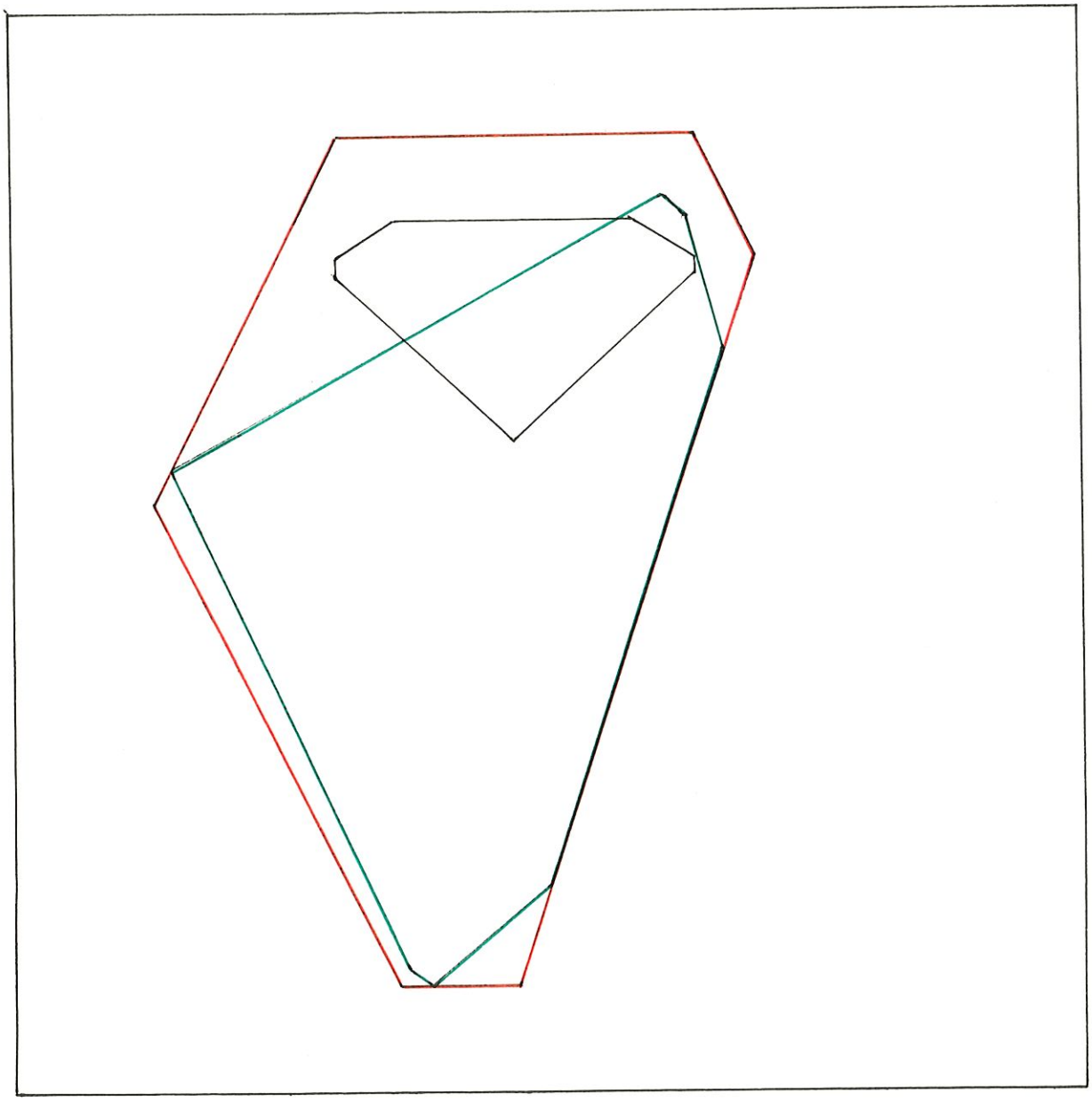
B. Le diamant de référence P_R (en noir) est déterminé par ses sommets :

1. (-1 , 0.3)
2. (1 , 0.3)
3. (1.5, 0)
4. (1.5,-0.1)
5. (0 ,-1.5)
6. (-1.5,-0.1)
7. (-1.5, 0)

C. Tableau des résultat

	Algorithme de Thoai	Algorithme modifié
Nombre d'itérations		15
Nombre de contraintes		15
$V(S_{final})$	* * *	74
$V(S_{max})$	* * *	75
Temps CPU (sec.)		1.29

Dans ce deuxième test, le nombre de sommet final dépasse 25000 pour la méthode de Thoai avant que la solution "complète" du problème ne soit trouvée. En effet, comme nous l'avons signalé en débutant cette section, les variables u,v,p et q sont correctes mais la variable t reste à ajuster. Par contre l'algorithme que nous avons modifié conduit à la solution complète assez rapidement.



DILATATION = 2,26

TEST 3

A. Le diamant brut P_0 (en rouge) est défini par les inégalités suivantes :

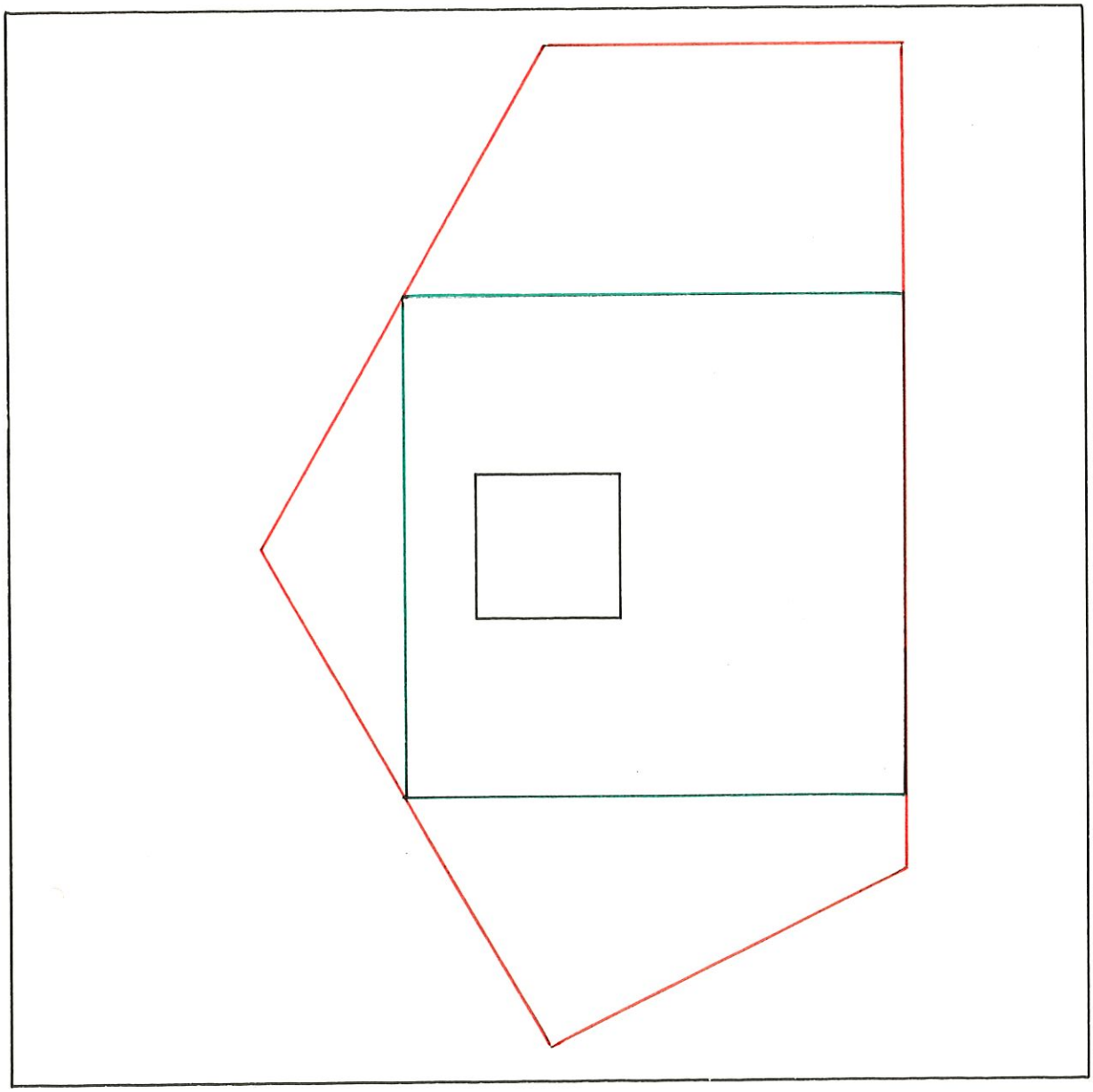
1. $-1.75x + y - 7 \leq 0$
2. $y - 7 \leq 0$
3. $x - 5 \leq 0$
4. $0.5x - y - 7 \leq 0$
5. $-1.75x - y - 7 \leq 0$

B. Le diamant de référence P_R (en noir) est déterminé par ses sommets :

1. (1, 1)
2. (1,-1)
3. (-1,-1)
4. (-1, 1)

C. Tableau des résultats :

	Algorithme de Thoai	Algorithme modifié
Nombre d'itérations	24	15
Nombre de contraintes	29	15
$V(S_{final})$	10282	48
$V(S_{max})$	10282	50
Temps CPU (sec.)	232.07	1.25



DILATATION = 3,5

TEST 4

A. Le diamant brut P_0 (en rouge) est défini par les inégalités suivantes :

1. $-4x + 11y - 202 \leq 0$
2. $y - 18 \leq 0$
3. $x + y - 22 \leq 0$
4. $7x + 4y - 115 \leq 0$
5. $x - 13 \leq 0$
6. $7x - 4y - 91 \leq 0$
7. $7x - 6y - 105 \leq 0$
8. $-y - 14 \leq 0$
9. $-x - 3y - 44 \leq 0$
10. $-6x - 7y - 132 \leq 0$
11. $-7x - 2y - 117 \leq 0$
12. $-x - 17 \leq 0$
13. $-6x - 5y - 142 \leq 0$

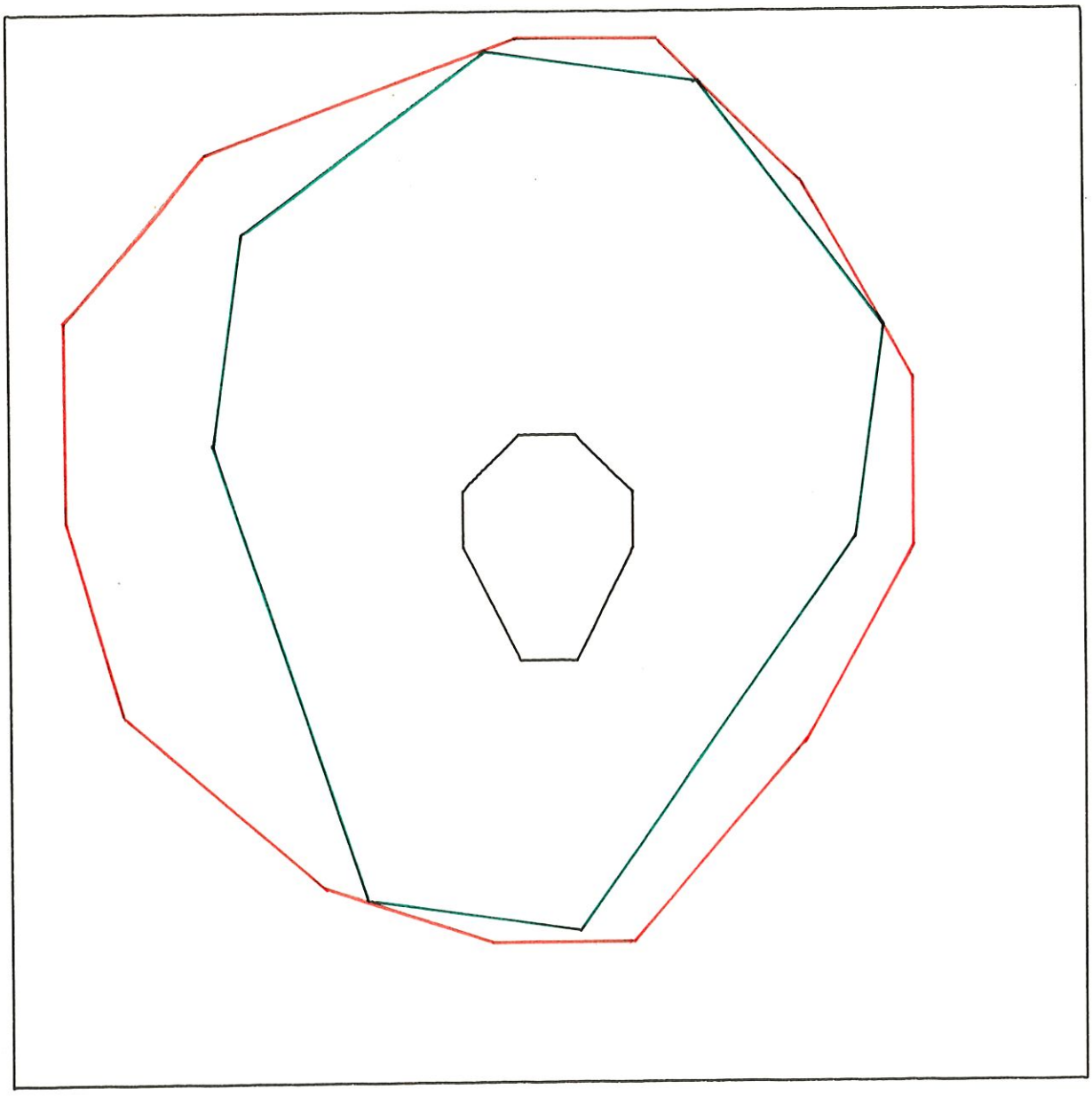
B. Le diamant de référence P_R (en noir) est déterminé par ses sommets :

1. (1, 4)
2. (3, 2)
3. (3, 0)
4. (1,-4)
5. (-1,-4)
6. (-3, 0)
7. (-3, 2)
8. (-1, 4)

C. Tableau des résultats :

	Algorithme de THOAI	Algorithme modifié
Nombre d'itérations		44
Nombre de contraintes		23
$V(S_{final})$		223
$V(S_{max})$		223
Temps CPU (sec.)		15.7

Dans ce test, comme dans celui qui va suivre, l'algorithme de Thoai ne trouve pas la solution (même u,v,p,q ne sont pas corrects), bien qu'il semble s'en rapprocher lentement. Ceci est dû à la taille des problèmes traités.



DILATATION = 3,81

TEST 5

A. Le diamant brut P_0 (en rouge) est défini par les inégalités suivantes :

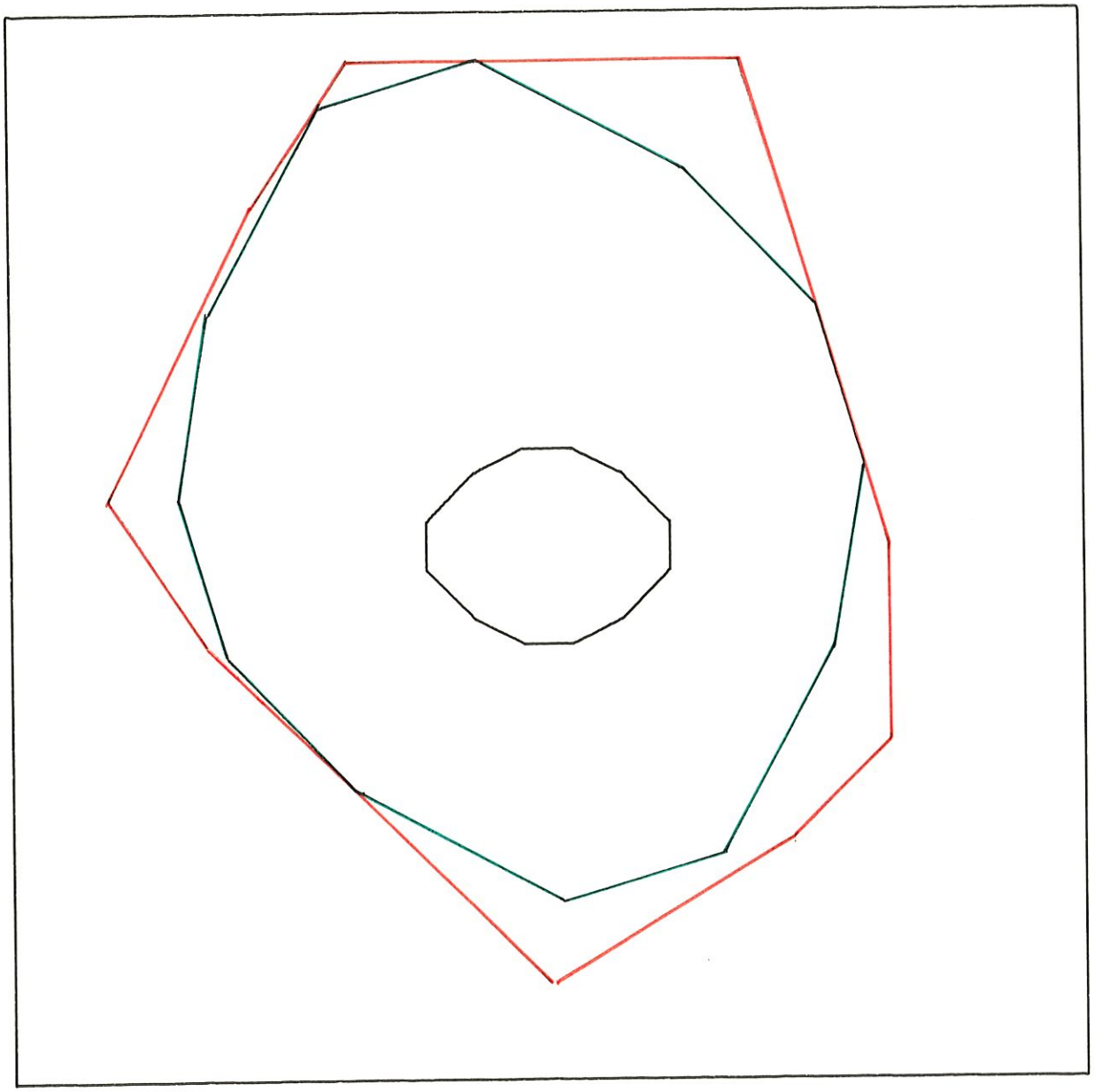
1. $y - 10 \leq 0$
2. $10x + 3y - 70 \leq 0$
3. $x - 7 \leq 0$
4. $x - y - 11 \leq 0$
5. $3x - 5y - 45 \leq 0$
6. $-x - y - 9 \leq 0$
7. $-3x - 2y - 25 \leq 0$
8. $-2x + y - 19 \leq 0$
9. $-3x + 2y - 32 \leq 0$

B. Le diamant de référence P_R (en noir) est déterminé par ses sommets :

1. (0.5 , 2)
2. (1.5 , 1.5)
3. (2.5 , 0.5)
4. (2.5 ,-0.5)
5. (1.5 ,-1.5)
6. (0.5 , -2)
7. (-0.5 , -2)
8. (-1.5 ,-1.5)
9. (-2.5 ,-0.5)
10. (-1.5 , 1.5)
11. (-0.5 , 2)

C. Tableau des résultats :

	Algorithme de THOAI	Algorithme modifié
Nombre d'itérations		38
Nombre de contraintes		35
$V(S_{final})$		228
$V(S_{max})$		228
Temps CPU (sec.)		34.4



DILATATION = 3.41

TEST 6

A. Le diamant brut P_0 (en rouge) est défini par les inégalités suivantes :

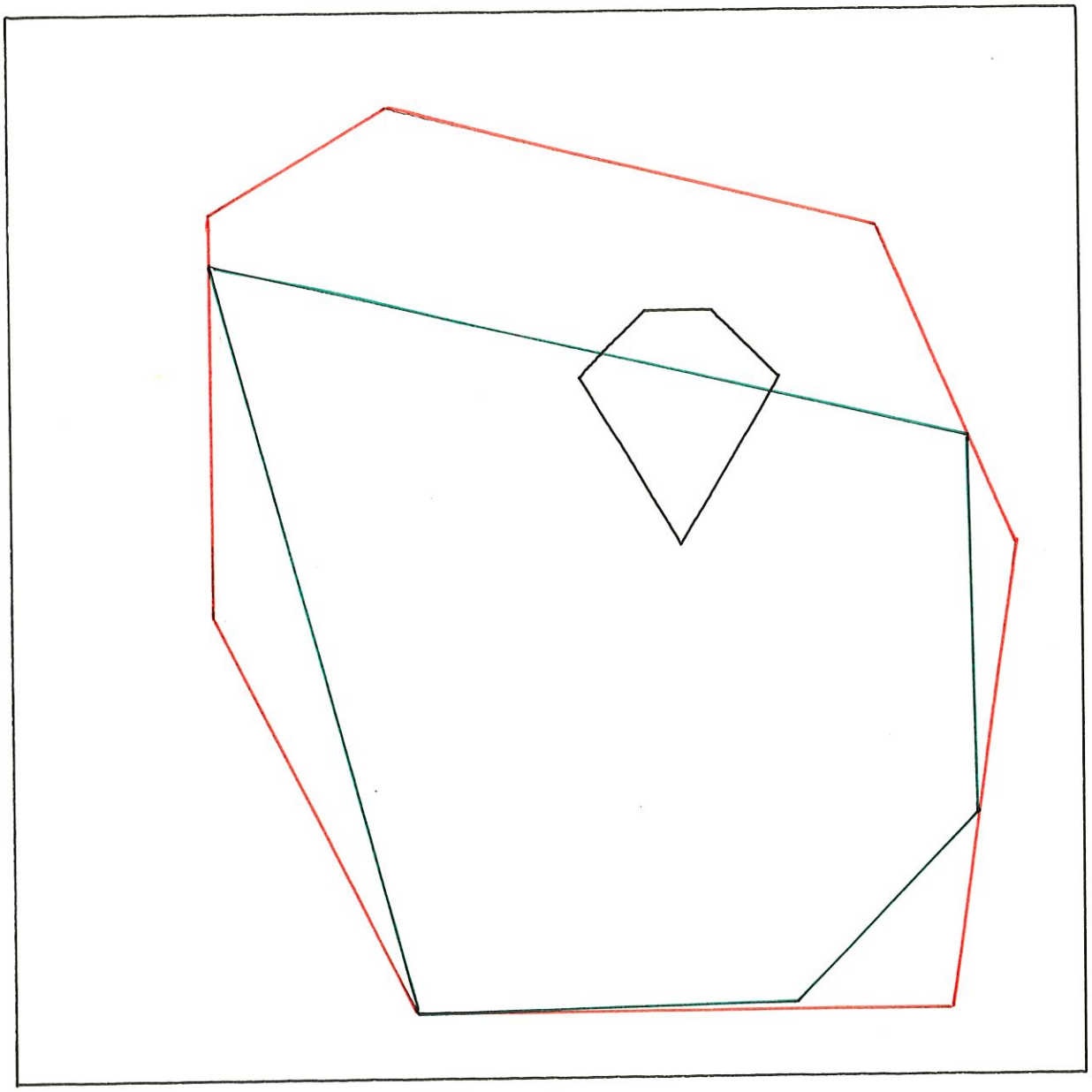
1. $-y - 7 \leq 0$
2. $-2x - y - 11 \leq 0$
3. $-x - 5 \leq 0$
4. $-3x + 5y - 40 \leq 0$
5. $0.25x + y - 6 \leq 0$
6. $7x + 3y - 49 \leq 0$
7. $7x - y - 49 \leq 0$

B. Le diamant de référence P_R (en noir) est déterminé par ses sommets :

1. (2 , 0)
2. (0.5, 2.5)
3. (1.5, 3.5)
4. (2.5, 3.5)
5. (3.5, 2.5)

C. Tableau des résultats :

	Algorithme de THOAI	Algorithme modifié
Nombre d'itérations		28
Nombre de contraintes		23
$V(S_{final})$		117
$V(S_{max})$		117
Temps CPU (sec.)		6.64



DILATATION = 3.99

CONCLUSION.

Le fait d'ajouter une seule contrainte anticonvexe à un programme convexe ordinaire transforme celui-ci en un problème relativement complexe. Nous avons dans ce mémoire étudié l'algorithme de Tuy permettant de résoudre un tel programme dit d.c. convexe. L'algorithme implémentable découlant de cette approche n'était pas très performant mais les modifications que nous avons apportées ont permis d'accélerer nettement sa vitesse de convergence. Grâce à ces améliorations nous avons pu résoudre des problèmes de taille plus importante.

Cependant, pour le lecteur interessé, il reste quelques domaines de recherche. Dans l'élaboration du calcul des nouveaux sommets, une étude plus appronfondie de la précision numérique nous paraît utile. D'autre part, en vue d'obtenir de meilleures coupes, on pourrait imaginer différentes façons de mettre à profit l'information locale dont on dispose.

REFERENCES.

[1] Falk J. and Hoffman K.R, "A successive underestimation method for concave minimization problems ", Math.Oper.Res.1, 251- 259, 1976.

[2] Hiriart-Urruty J.- B, " Generalized differentiability, duality and optimization for problems dealing with differences of convex functions ", Rapport interne, université de Toulouse, 1986.

[3] Horst R., de Vries J., Thoai NG. V., " On finding new vertices and redundant contraints in cutting plane algorithms for global optimization ", Rapport interne, Université de Trêves, 1987.

[4] Rockafellar R.T., "Convex analysis ", Princeton University Press, N.J., 1970.

[5] Strodiot J.-J., Nguyen Van Hien, " Taille optimale de diamant: Méthode Diam1 ", Rapport technique n°79/5. Département Mathématique F.U.N.D.P. 1979.

[6] Thieu T.V., Tam B.T., Ban V.T., "An outer approximation method for globally minimizing a concave function over a compact convex set ", Acta Mathematica Vietnamica, Vol. 8, n°1, 21-40, 1983.

[7] Thoai NG.V., "A modified version of Tuy's method for solving d.c. programming problems", Rapport interne, Institue of mathematics, Hanoï et Core UCL, 1985.

[8] Tuy H., "Convex programs with an additional reverse convex constraint", J.O.T.A. Vol.52, n° 3., 1987.

[9] Tuy H., "A general deterministic approach to global optimization via d.c. programming", presented at the Fermat days: Mathematics for optimization, Toulouse, 1985.

[10] Tuy H.," On outer approximation methods for solving concave minimization problems,Acta Mathematica Vietnamica, Vol. 8, 3-34,1983.